AFRL-RY-WP-TR-2015-0174

NONPARAMETRIC REPRESENTATIONS FOR INTEGRATED INFERENCE, CONTROL, AND SENSING

John Fisher and Jon How
Massachusetts Institute of Technology

Trevor Darrell
ICSI

Luis Galup
BAE Systems

Andreas Krause
ETH

Stefano Soatto
University of California Los Angeles

NOVEMBER 2015
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
November 2015	Final	15 September 2011 – 30 May 2015

4. TITLE AND SUBTITLE
NONPARAMETRIC REPRESENTATIONS FOR INTEGRATED INFERENCE, CONTROL, AND SENSING

5a. CONTRACT NUMBER
FA8650-11-1-7154

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61101E

6. AUTHOR(S)
John Fisher and Jon How (Massachusetts Institute of Technology)
Trevor Darrell (ICSI)
Luis Galup (BAE Systems)
Andreas Krause (ETH)
Stefano Soatto (University of California Los Angeles)

5d. PROJECT NUMBER
1000

5e. TASK NUMBER
11

5f. WORK UNIT NUMBER
Y015

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology
32 Vassar Street
32-D468
Cambridge, MA 02139

ICSI
BAE Systems
ETH
University of California Los Angeles

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

Defense Advanced
Research Projects Agency
DARPA/DSO
3701 N. Fairfax Dr.
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYAT

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TR-2015-0174

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. Report contains color. This material is based on research sponsored by Air Force Research Laboratory (AFRL) and the Defense Advanced Research Agency (DARPA) under agreement number FA8650-11-1-7154. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL) and the Defense Advanced Research Agency (DARPA) or the U.S. Government.

14. ABSTRACT
The objective of this research program was to develop mathematical foundations of information gathering through an integrated theory of sensing, inference, and control. The goal of the team was to develop a new framework for autonomous operations that will extend the state of the art in distributed learning and modeling from data, and tightly integrate these models into new decentralized cooperative planning algorithms. The main output of this effort will be a fundamental theory to integrate decentralized information driven planning methods for heterogenous teams with nonparametric Bayesian models of uncertainty. The feasibility and aspects of the value of the theory were demonstrated via integrated software and hardware experiments.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jared Culbertson
Unclassified	Unclassified	Unclassified	SAR	70	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

1 Objective

The objective of this research program was to develop mathematical foundations of information gathering through an integrated theory of sensing, inference, and control. The goal of the team was to develop a new framework for autonomous operations that will extend the state of the art in distributed learning and modeling from data, and tightly integrate these models into new decentralized cooperative planning algorithms. The main output of this effort will be a fundamental theory to integrate decentralized information driven planning methods for heterogenous teams with nonparametric Bayesian models of uncertainty. The feasibility and aspects of the value of the theory were demonstrated via integrated software and hardware experiments.

Phase I included an extensive set of mathematical and algortihmic developments which formed the basis of an integrated system. Bayesian inference represented by graphical models mediated between sensors and event probabilities of interest. Temporal Logic mediated between the use of graphical models for inference and the interpretation of system queries. In the proposed architecture, constructive Temporal Logic approach reduces first-order logic queries to a system of graphical models.

During, phase 2 algorithmic development emphasized transitioning from ensor–centric to scene–centric processing. As such, issues such as sensing geometry and the associated nuisance parameters, noisy and missing data, and mult-view and multi-modal sensing were important considerations for modeling and development. Methods to exploit information measures and their relation to the instantiated graphical structures were developed to investigate the trade off computational resource costs with the quality of approximate inference methods. Hierarchical Bayesian nonparametric methods were investigated for the purpose of modleing both contextual representations and specific instances of object, attributes and relations envisioned under the program.

While a significant aspect of MSEE Phase II and III was devoted system development, it is still the case that *fundamental research* in distributed planning and control, sensor and information management, and intent recognition were investigated to achieve the amitious goals of the program.

2 Overview

We provide an overview of the system developed by the MIT team as well as a description of the research results which are further detailed in technical publication listed at the end of this report.

2.1 Team Members

Table 1 lists the various key members of the team (by institution) and their primary areas of expertise and responsibilities.

Org	Capabilities & Responsibilities	Key Personnel
MIT	BNP Models, Inference, & Planning.	Dr. John Fisher, Prof. Jon How
ICSI	BNP Models, Large scale object recognition.	Prof. Trevor Darrell
UCLA	3D/Geometric scene representation	Prof. Stefano Soatto
ETH Zurich	Discrete and mixed integer-continuous optimization.	Prof. Andreas Krause
BAE Systems	Temporal logic & system integration	Dr. Luis Galup, Ms. Wendy Mungovan, Mr. Manuel Cuevas

Table 1: Team members and primary technical expertise. Note that Prof. Krause joined the team at the beginning of Phase 2, while Dr. Galup and Ms. Mungovan left the team at the completion of Phase 2.

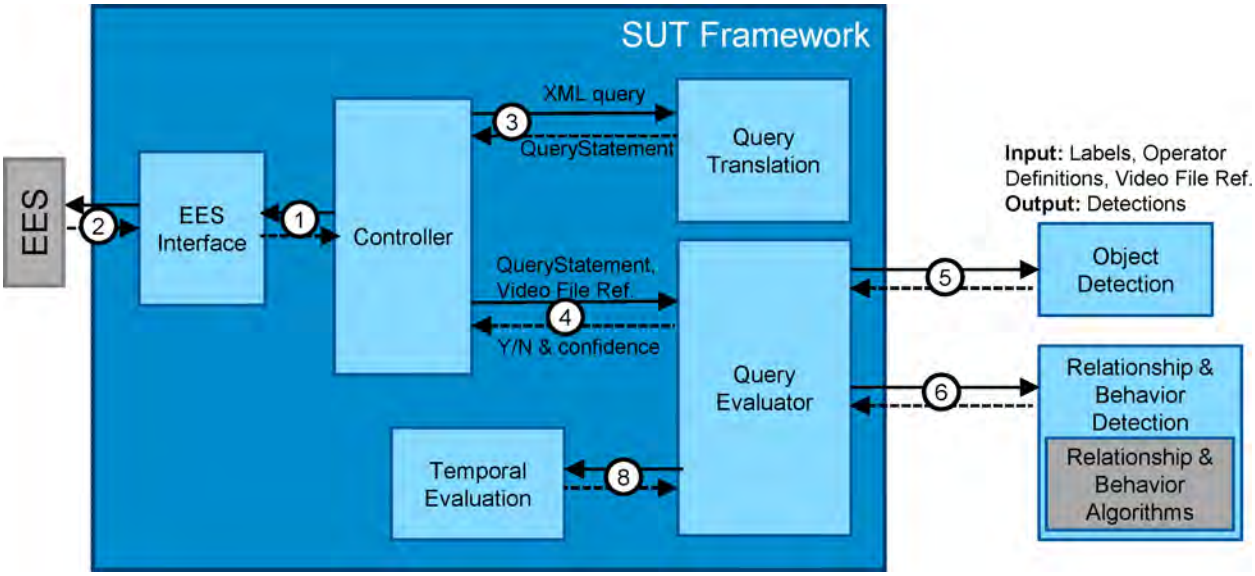


Figure 1: Function system block diagram of MIT MSEE SUT implementation.

2.2 System Description

Figure 1 depicts the funtional system block diagram of the MIT MSEE SUT implementation. Communication with the EES, query ingestion, query parsing, and predicate tasking are performed within the SUT Framework developed by BAE. Scene modeling including labeling of moving and static objects as well as determining 3D geometry are performed off-line and stored in a database. Geometric modeling is performed by modules developed by UCLA while object tracking and scene labeling are performed by modules developed by MIT. Finally, object labeling (including tracked objects) are also performed off-line using a variant of Caffe developed by ICSI. All results are stored in postgres databases for later indexing during query processing.

The goal was to develop a working system for query-based scene understanding that integrates physical sensor models of video cameras, Bayesian reasoning via structured graphical models and integration of contextual models. Following the Phase 2 demonstration, the team had produced a functioning end-to-end system demonstrating the following functionality:

- large scale object classification,
- semi-automated 3D scene modeling,
- extensible system for predicate implementation,
- ability to reason over geometric, dynamic, and behavioral relations

The Phase 2 system emphasized sensor-centric processing for predicate reasoning with extensions to 3D reasoning aided by 3D scene representation. An intial working version of the system was transitioned to Air Force Research Laboratory. Recent extensions are in the process of being transitioned, as well.

2.3 Processing Flow

Figure 2 depicts the conceptual approach of the MIT MSEE design. Here, an intermediate representation comprised of – (1) a scene represenation, (2) object and mover attribution, and (3) tracking of movers – separates sensing from reasoning. The advantage is that reasoning can be defined in terms physical relations (as paramerized by the representation) and logical functions. Queries (as prescribed by the formal language specification) are comprised of predicates which are defined deterministically over the intermediate representation. As such, uncertainty is modeled in the intermediate representation (e.g. due to sensor noise

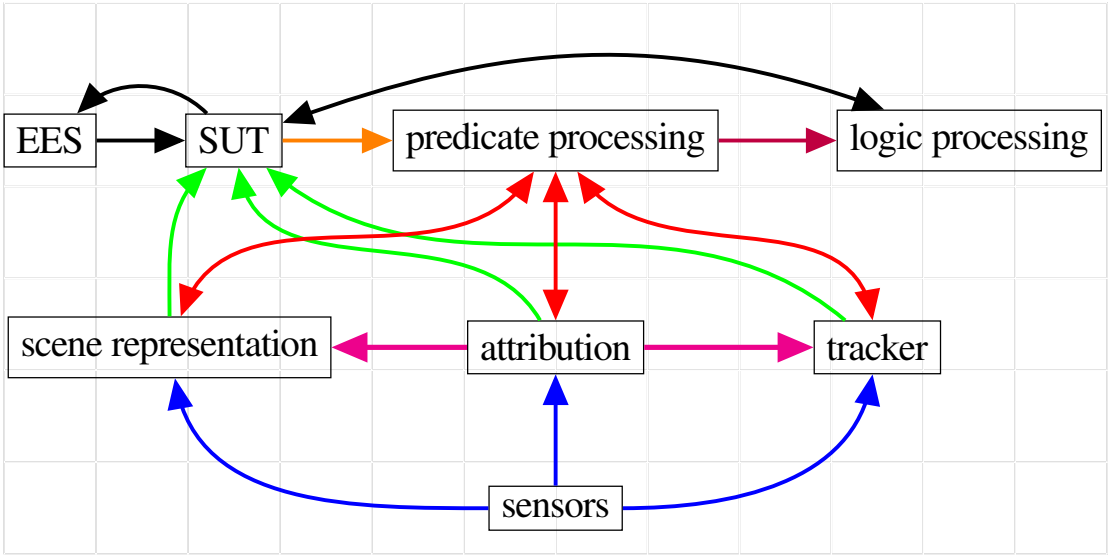


Figure 2: Conceptual Diagram of MIT MSEE Design

and model mismatch) rather than in the reasoning system.

As a result, predicates are mapped to collections of inference algorithms implemented as modular and composable probabilistic graphical models. Conceptually one could instantiate a monolithic model and focus inference on the relevant latent variables, however, for the complexity of the scenes contemplated by MSEE and the number of sensors, such an approach is intractable. An additional (and substantial) benefit of the modular approach is that it allows efficient and principled handling of nuisance parameters only when necessary, optimization of the measurement process, as well as instantiation of only those aspects of the representation that are relevant to the query. The modular approach also easily lends itself to parallelization.

We note that graphical models are not a panacea, rather they are a framework. Whle they aid in organizing relationships between queries, sensors, and the scene while making dependency assumptions explicit, they only *suggest* methods for inference. The critical choice of how to perform inference in a given graphical model is left to the designer and will depend on the definitions of predicates which reason over that graphical model. That being said, the modular approach allows these models to be designed independently.

3 System Performance

3.1 Predicate Handling Framework

Predicate analysis and evaluation are implemented as a separate module (denoted by the red box in Figure 3. In the MIT design and implementation, predicates results are treated as independent. This choice was made for practical reasons due to the fact that modeling (and reasoning) over *dependent* predicates is not feasible given the number of relations the system would have to consider. Treating them independently is akin to making what is known at the naive Bayes assumption. One practical consequence is that predicates can be evaluated in *parallel* allowing for significant speedups in analysis. Predicates are roughly grouped into three categories, *behavior* predicates, *relationship* predicates, and *action* predicates. These groupings are shown in Table 2.

As currently implemented, incorporation of new predicates is a straightforward process of defining the predicate as a logical function of its inputs and their relation to the physical properties of the scene. For example, the predicate “together” is defined in terms of the proximity of the arguments specified in physical units (when available) or in terms of sensor dimensions (e.g. pixels) when the physical units are not availalbe.

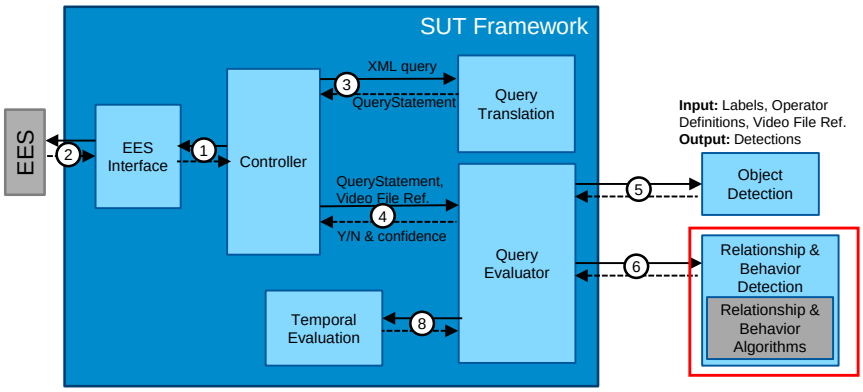


Figure 3: Predicate evaluation is implemented as a separate module, denoted by red box in figure.

Table 2: Predicate categorization and implementation status.

Behavior		Relationships		Actions	
Implemented	*Not Implemented*	*Implemented*	*Not Implemented*	*Implemented*	*Not Implemented*
1. Starting		1. Same-object	1. Touching	1. Driving	1. Loading
2. Moving		2. Part-of	2. Facing	2. Entering	2. Unloading
3. Stopping		3. CLOS	3. Facing-opposite	3. Exiting	3. Donning
4. Stationary		4. Occluding	4. Inside	4. Crossing	4. Doffing
5. Turning		5. On	5. Outside	5. Carrying	5. Wearing
6. Turning-right		6. Together	6. Putting-in	6. Mounting	6. Swinging
7. Turning-left		7. Closer		7. Dismounting	
8. U-turn		8. Father		8. Putting-up	
9. Crawling		9. Below		9. Taking-down	
10. Walking		10. Same-motion		10. Throwing	
11. Running		11. Opposite-motion		11. Catching	
12. Sitting		12. Following		12. Putting-down	
13. Standing		13. passing		13. Picking-up	
14. Talking				14. Dropping	
15. Writing					
16. Reading					
17. Eating					
18. Pointing					
19. Open					
20. Closed					

The former is always possible so long as the scene properties have been specified (described elsewhere) in which case the predicate makes use of so-called "helper functions" used to define the relation of predicate arguments to the scene being analyzed. Whether to utilize the physical dimensions of the scene (which is subject to sensor uncertainty) and the associated helper functions is left to the predicate designer.

Details of the predicate handling framework are shown in the system block diagram of Figure 4. The predicate handling framework (1) interfaces with the MSEE framework (i.e the system which receives the query from the EES and parses it, (2) accesses the database of precomputed analysis (tracks of movers, labels of objects, and the geometric description of the scene), (3) determines the order and combination of which predicates to evaluate, and (4) handles various special cases and checks for errors.

The syntax for the MSEE framework call to the predicate handling framework (circle 1 in Figure 4) is shown in table 3. Having received the predicate call from the MSEE framework, the predicate handling framework separate predicate calls for each valid combination of unary, binary, or ternary arguments along with associated track and scene info. The syntax for calling a specific instance of a predicate (circle 2 in Figure 4) is shown in Table 4. While the MSEE framework can parallelize calls to the predicate handling

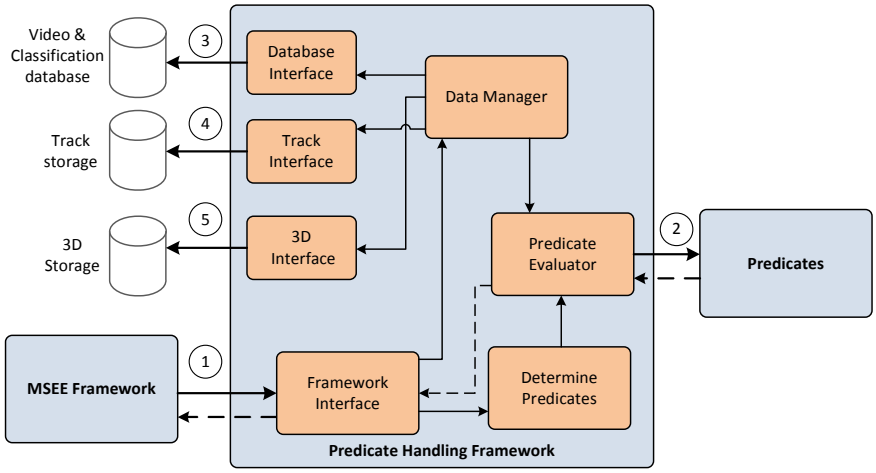


Figure 4: Predicate handling framework. Predicates access the results of sensor data processing via a database of pre-computed analysis including 3D Scene analysis, tracking of moving objects, and classification of moving and static objects.

Table 3: Syntax for MSEE framework call to predicate handling framework

Syntax:

```
poe.MIT(<predicate>,<time window start>,<time window end>,
                                        <obj1>,[obj2],[obj3]);
```

Inputs:	**Output:**
Predicate to evaluate (string).	Matrix containing probability of the predicate being true for each of the object combinations.
Time Window of interest (string).	
List of objects for each of the predicate inputs (videoID, trackID).	

framework (once the query is parsed) across predicates, further parallelization is possible within the predicate handling framework across instances of argument combinations.

3.2 Predicate Processing Time:

Figures 5 and 6 provide details of the processing time broken down by predicate. Recall that tracking, scene construction, and object labeling are peformed ahead of any query time. Consequently, the values in these figures reflect the time the complete predicate reasoning and data base access times and *do not* include sensor processing time. In future implementations, it would be straightforward to store sensor processing time as part of the pre-processing step. This would allow analysis that computes both sensor processing time and logic processing time. Both depend on the complexity of the query, the complexity of the scene, the number of sensors, the time duration over which the query is applied.

Figure 5 reflects the total time to process each predicate for a given query. For a given query, this would be the time to process all valid arguments for a specific predicate. As seen in the figure, most predicates take very little time to process. Multiple values for a given predicate reflect that the predicate was used in more than one query. The differences in processing time are a consequence of the number of arguments

Table 4: Syntax for predicate handling framework to individual predicate instances.

Syntax:

`predicate_ptr(info,objs,tracks, scene_3d,params);`

Inputs:	**Output:**
General Info (cell array)	Structure containing indicator whether predicate is true or false, and associated probability.
Track Instances (cell array).	
Tracks (cell array).	
3D Representation (function pointers).	
Predicate parameters (structure)	

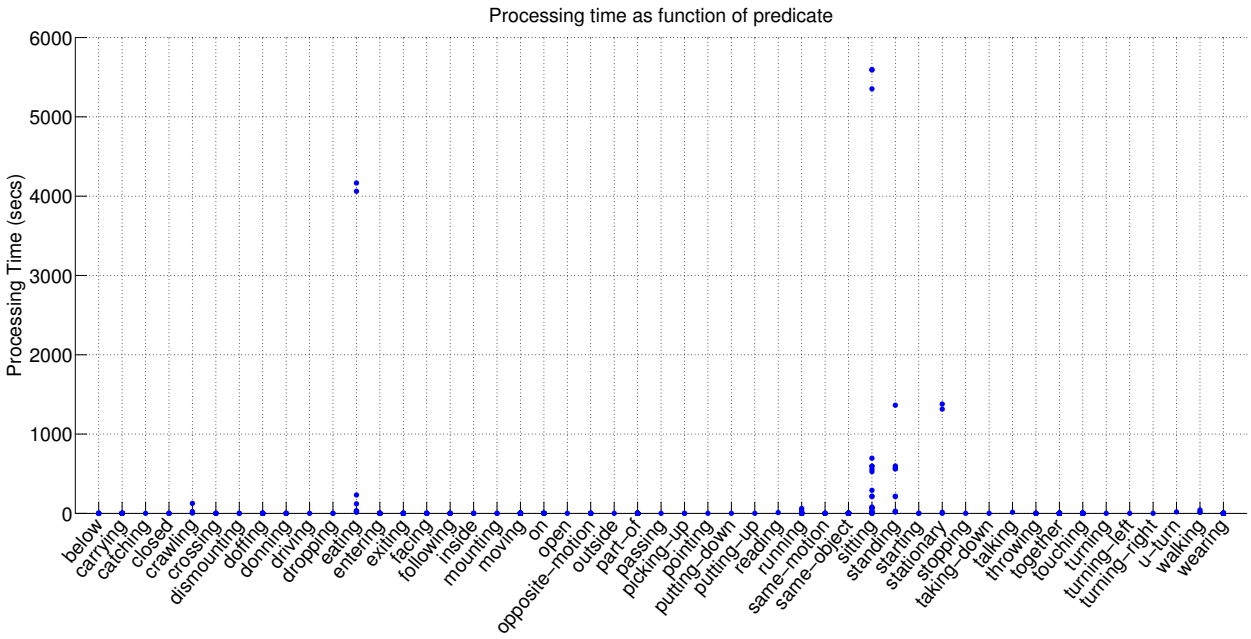


Figure 5: Processing Time as a Function of Predicates

passed to the predicate for that particular query. These values are more reflective of the complexity of the various queries used for Phase 2 testing. Figure 6 reflects the time to process each predicate for a single instance. Here the differences in processing time are reflective of the temporal duration associated with the particular instance of the predicate evaluation.

3.3 Query Accuracy:

Phase 2 involved 276 queries submitted to the query. Of those, 218 queries were processed. Some predicates were not supported and, as a result, any query which contained those predicates was not processed (a total of 58). The 218 processed queries resulted in 390 predicate calls. This is indicative of the fact that many queries were comprised of a single predicate and very few queries incorporated 4 or more predicates (see Figure 8(left))

The system performance for 218 queries is detailed in Figure 7. The table at the left of the figure provides counts of true positives, false positives, true negatives, and false negatives. The chart at the right depicts the relative percentages. We note that the system as implemented has a bias towards returning a "true" value.

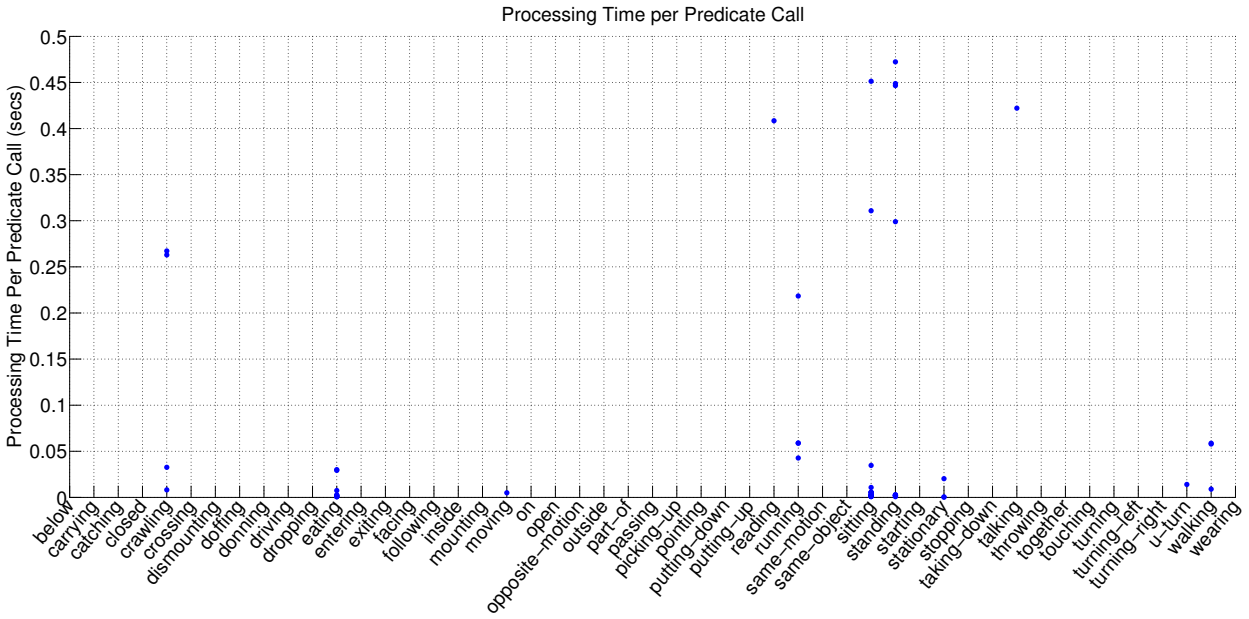


Figure 6: Processing Time Normalized by Predicate Evaluations

		Provided	
		True	*False*
Correct	*True*	77	46
	False	51	44

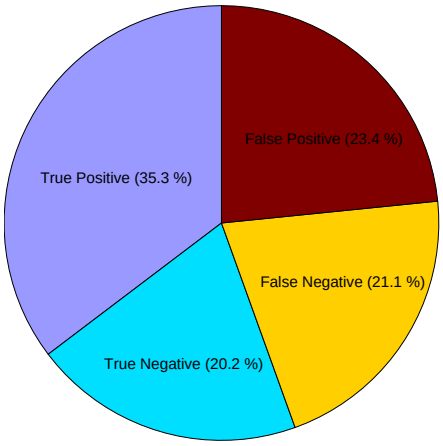


Figure 7: Query Performance.

This is due to interpreting a query (or predicate) as being true for a given time period even if it is true only once (i.e. at a single point in time). The consequence is that as the time period grows, even if a predicate reports a low-probability of being true at every time instance, the overall probability approaches unity as the length of the time period grows. This is perhaps the simplest interpretation of what constitutes a query or predicate being true. Other approaches could be adopted, but were not investigated.

Not all predicates are equal: While the figure 7 reflects average performance for the system when evaluated over the choice of queries for phase 2, it is unlikely that it accurately reflects the overall system performance as the queries chosen for testing were biased towards the use of a small number of predicates.

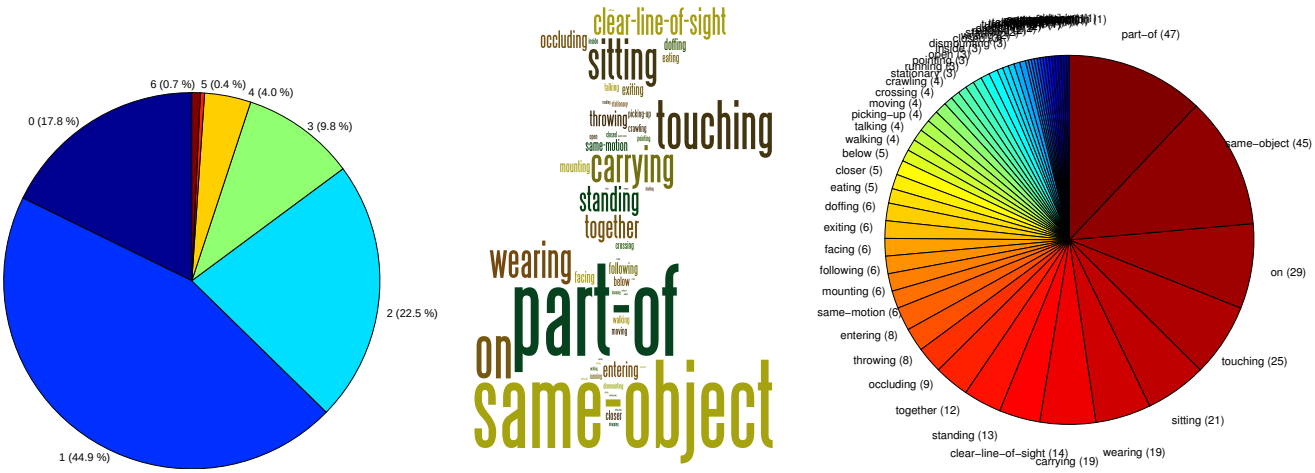


Figure 8: (left) Breakdown of number of predicates per query, (middle) wordle where the size of the predicate name reflects the usage frequency across queries, and (right) pie-chart with counts of predicate usage across queries.

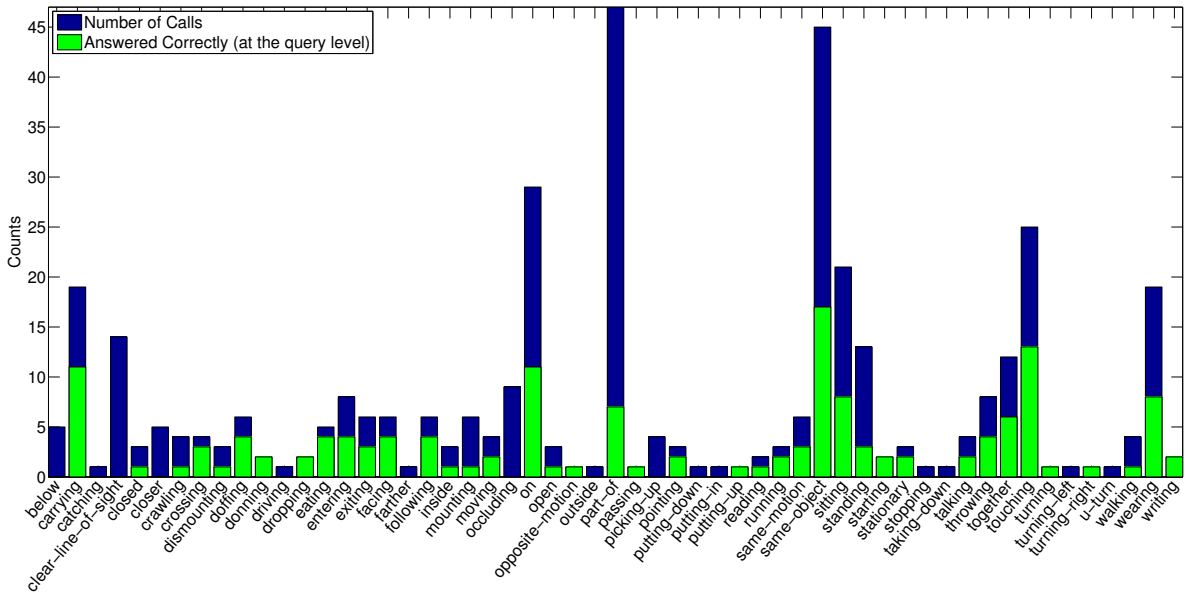


Figure 9: Accuracy of each predicate across all queries.

This can be seen in Figure 8 which visualize the relative frequency with which predicates were used within the phase 2 testing queries. As can be seen, "part-of" and "same-object" were called significantly more often, 47 and 45 times each, as compared to "together" which was called once. Consequently, the query performance numbers are largely reflective of the performance on the most frequently called predicates. Whether this is an accurate reflection depends on the anticipated scenarios in which such a system would be used.

Figure 9 shows the relative accuracy of each predicate where blue reflects the number of times the predicate was called and green the number of times the predicate returned a correct answer. We note that performance on many queries is substantially above guessing, however, on three of the most frequently called predicates, "on", "part-of", and "same-object", the performance is fairly poor resulting in a larger impact on system performance.

4 Preprocessing

4.1 3D Scene Modeling

The methodology for constructing 3D information of the scene is described in the material provided in this section. It was noted during the course of the program that the quality of the reconstruction, upon which accurate spatial reasoning depends, is impacted by both the accuracy of the intrinsic parameters of the cameras and knowledge of the sensing geometry. The former was provided, but the latter was not. Consequently, state-of-the-art methods employing automatic detection of correspondences were utilized. The accuracy of these methods depends greatly on both the sensor geometry and the content of the scene. For some of the scenes, these were not adequate to yield acceptable performance and as a result, manual correspondences were needed.

Details of the methodology are found in Section 8.1.

4.2 Boundary Accurate Tracker

As part of pre-processing the MIT design tracks all movers, storing the results in a database. Both the location (within the sensor view) and the boundary of the object are computed. The tracker was partially developed under the MSEE program and implements layered tracking, adaptive appearance models, and occlusion reasoning.

Details of the methodology are found in Section 8.2.

4.3 Object Classification

As part of pre-processing the MIT all movers and static objects are classified using a variant of Caffe adapted to the MSEE object hierarchy. The implementation provided by ICSI (co-PI Darrell) did not fully implement the hierarchy, but nevertheless provided reasonable performance on many objects of interest. One impact on performance is that only the highest scoring class was maintained for each object. As such, errors in the use of the classfier had an undue impact on system performance. More robust performance would be obtained if a full or partial distribution were maintained as part of the pre-processing. This is feasible, but would complicate query processing owing to the increased combinatorial complexity. Consequently, the simple approach was chosen for phase 2.

Details of the methodology are found in Section 8.3.

5 Discussion

5.1 Scene-wide 3D reasoning requires significant prior knowledge of sensor placement.

As described in the formal language specification, queries and associated predicates were defined as reasoning over a scene rather than a sensor. That is, the collection of sensors provides observations of the scene, but the scene itself may not be limited to field-of-view of the sensors. Additionally, many predicates (as defined) require extended spatial and temporal reasoning. For example, the predicate "clear-line-of-sight" can potentially be used to reason over persons (or locations) that are not visible in the same sensor. Furthermore, it is entirely possible that one could be interested in processing this particular predicate in order to reason over individuals who may have at one time been visible in different sensors, but at the time of query one or both individuals may no longer be directly observed. This does **not** preclude processing the query. As part of scene understanding, individuals are tracked and as such, even when not directly observed, the system has some information as to their location. While the example is somewhat extreme, it highlights the fact that, as defined, reasoning over the 3D geometry of the scene is unavoidable **unless** one knows in advance such queries will not be utilized. Many predicates implicitly require this capability.

The importance of this discussion is that it underlies the critical need for knowledge of the sensing geometry. In the absence of this information, it must be inferred. In many cases for the Phase 2 testing, the information was not adequately provided. For example, camera locations were (roughly) provided, but direction of viewing was not. Furthermore, state-of-the-art methods for finding correspondences also

proved to be inadequate for inferring the scene geometry to an acceptable quality for purposes of processing queries. As a consequence, a manual and labor-intensive process was necessary in order to accommodate the potential for these queries. The performers had no way of knowing ahead of time whether testing queries would require this level of reasoning. It is the opinion of the PI that this complication was unnecessary and did not serve the goals of the program.

5.2 Significant tradeoffs for state-of-the-art video-based object tracking.

Many of the predicates, especially those involving gestures or actions, require some segmentation of the of the body pose. Consequently, this project chose to implement a video tracking algorithm which produced accurate object boundaries. While results were satisfactory, real-time performance is challenged by current computating capabilities. As such, tracking speed was on the order of 10-20 seconds per frame. Some gains may be achieved by better utilzation of multi-core processors and/or gpu processing. However, in the current framework, object tracking is performed off-line in order to focus on reasoning performance. There exist video trackers which are capable of tracking objects in real-time, however, these trackers do not produce boundary-accurate results and furthermore, do not perform well when the number of moving objects is greater than ten.

This issue might be mitigated by combining fast bounding box trackers densely and boundary accurate trackers only when the query requires it. Implementation of such a scheme was entertained in the original design, but it was felt that the added complexity would risk successful completion of a working system.

5.3 Rolling shutter effect significantly degrade moving camera analysis.

For moving camera data, correspondences across frames were both dense and fairly robust. However, rolling shutter artifacts, which manifest themselves as the image appearing to warp from frame-to-frame, result in state-of-the-art structure-from-motion algorithms generating severely degraded results. While one could incorporate rolling shutter into the model, to do so was beyond the scope of this project.

6 Students

The following is a list of students that have been supported by the project listed by institution.

6.1 MIT

- Randi Cabezas – PhD student (due to graduate Summer 2016)
- Jason Chang – Completed PhD, now at Google
- Zoran Dzunic – PhD student (due to graduate Fall 2015)
- Oren Freifeld – Postdoc
- Dan Levine – Completed PhD student, now at Jet Propulsion Laboratory
- Dahua Lin – Completed PhD students, now professor at CUHK
- Guy Rosman – Postdoc

6.2 UCLA

- Avinash Ravichandran - Completed postdoc; now at Amazon, INC.
- Jonathan Balzer - Completed postdoc; now at Vathos, GmbH (co-founder)
- Timothy Brightbill - Completed undergraduate degree
- Joshua Hernandez PhD student (due to graduate Summer 2015)
- Vasiliy Karasev PhD student (due to graduate Summer 2015)
- Nikolaos Karianakis - PhD student
- Sim-Lin Lau Staff Researcher Associate
- Stephen Phillips - Completed undergraduate degree
- Siyang Tang Completed MS degree; now at Apple, INC.
- Brian Taylor PhD student (due to graduate Fall 2015)

- Chaohui Wang Completed postdoc; now at Max Planck Institute

6.3 ETH

- Yuxin Chen – PhD student (due to graduate Summer 2016)

6.4 ICSI

- Jiashi Feng – Postdoc
- Eric Tzeng – PhD student
- Ross Girshick – PhD student

7 Publications

During the course of this project, the PI and co-PIs published 40 conference and journal in a variety of relevant and diverse topics including Bayesian nonparemetric models, system control, object recognition, distributed sensing, Bayesian inference, tracking. A full list of project-related publications is maintained at the following URL

http://projects.csail.mit.edu/csail-msee/pubs.html

The following is a list of publications funded (or partially funded) by this project that have either appeared in the scientific literature (or are pending review).

List of Project Publications

[1] Jason Chang and John W. Fisher III. Efficient mcmc sampling with implicit shape representations. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, June 2011.

[2] Jason Chang and John W. Fisher III. Efficient topology-controlled sampling of implicit shapes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Sept 2012.

[3] J. Hernandez, N. Karianakis, and S. Soatto. Information driven exploration using poisson sampling over ising marginals. In *In preparation*, June 1 2012.

[4] Ke Jiang, Brian Kulis, and Michael Jordan. Small-variance asymptotics for exponential family dirichlet process mixture models. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3167–3175, 2012.

[5] Vasiliy Karasev, Alessandro Chiuso, and Stefano Soatto. Controlled recognition bounds for visual learning and exploration. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2924–2932, 2012.

[6] Sergey Karayev, Tobias Baumgartner, Mario Fritz, and Trevor Darrell. Timely object recognition. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 899–907, 2012.

[7] Brian Kulis and Michael Jordan. Revisiting k-means: New algorithms via bayesian nonparametric. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[8] Dahua Lin and John W. Fisher III. Coupling nonparametric mixtures via latent dirichlet processes. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 55–63, 2012.

[9] Dahua Lin and John W. Fisher III. Efficient sampling from combinatorial space via bridging. In Neil Lawrence and Mark Girolami, editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 694–702, April 2012.

[10] Dahua Lin and John W. Fisher III. Low level vision via switchable markov random fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439, June 2012.

[11] Dahua Lin and John W. Fisher III. Manifold guided composite of markov random fields for image modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2176–2183, June 2012.

[12] Georgios Papachristoudis and John W. Fisher III. Theoretical guarantees on penalized information gathering. In *Proc. IEEE Workshop on Statistical Signal Processing*, August 2012.

[13] Donglai Wei, Dahua Lin, and John W. Fisher III. Learning deformations with parallel transport. In *Proceedings of the 12th European conference on Computer Vision - Volume Part II*, ECCV'12, pages 287–300, Berlin, Heidelberg, 2012. Springer-Verlag.

[14] Randi Cabezas. Aerial reconstructions via probabilistic data fusion. S.m. thesis, Massachusetts Institute of Technology, 2013.

[15] Jason Chang and John W. Fisher, III. Topology-constrained layered tracking with latent flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec 2013.

[16] Jason Chang and John W. Fisher III. Object tracking with topology constraints and gaussian process flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec 2013.

[17] Jason Chang and John W. Fisher III. Parallel sampling of dp mixture models using sub-cluster splits. In *Advances in Neural Information Processing Systems 26*, pages 620–628. Dec 2013.

[18] Jason Chang, Donglai Wei, and John W. Fisher III. A video representation using temporal superpixels. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[19] Yuxin Chen and Andreas Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *International Conference on Machine Learning (ICML)*, 2013.

[20] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[21] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[22] Judy Hoffman, Eric Tzeng, Jeff Donahue, Yangqing Jia, Kate Saenko, and Trevor Darrell. One-shot adaptation of supervised deep convolutional models. *CoRR*, abs/1312.6204, 2013.

[23] D. Levine and J. P. How. Sensor selection in high-dimensional gaussian trees with nuisances. In *Proc. Neural Information and Processing Systems (NIPS)*, 2013. to appear.

[24] D. Levine, B. Luders, and J. P. How. Information-theoretic motion planning for constrained sensor networks. *J. Aerospace Information Systems (JAIS)*, 2013. to appear.

[25] Y. Zeng, C. Wang, Stefano Soatto, and S.-T. Yau. Nonlinearly constrained mrfs: Exploring the intrinsic dimensions of higher-order cliques. June 2013.

[26] Ning Zhang, Ryan Farrell, Forrest Iandola, and Trevor Darrell. Deformable Part Descriptors for Fine-grained Recognition and Attribute Prediction. In *ICCV*, 2013.

[27] J Chang and J. W. Fisher III. Parallel sampling of hdps using sub-cluster splits. In *Proceedings of the Neural Information Processing Systems (NIPS)*, Dec 2014.

[28] Jason Chang, Randi Cabezas, and John W. Fisher III. Bayesian nonparametric intrinsic image decomposition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Sept 2014.

[29] Yuxin Chen, Hiroaki Shioi, Cesar Fuentes Montesinos, Lian Pin Koh, Serge Wich, and Andreas Krause. Active detection via adaptive submodularity. In *Proc. International Conference on Machine Learning (ICML)*, 2014.

[30] Zoran Dzunic and John Fisher III. Bayesian Switching Interaction Analysis Under Uncertainty. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 220–228, 2014.

[31] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[32] Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains, 2014.

[33] Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, James Andrew Bagnell, and Siddhartha Srinivasa. Near-optimal bayesian active learning for decision making. In *To appear in Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.

[34] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio

Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[35] Sergey Karayev, Mario Fritz, and Trevor Darrell. Anytime Recognition of Objects and Scenes. In *CVPR*, 2014.

[36] Julian Straub, Guy Rosman, Oren Freifeld, John J. Leonard, and John W. Fisher III. A Mixture of Manhattan Frames: Beyond the Manhattan World. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[37] Yuxin Chen, S. Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Proc. International Conference on Learning Theory (COLT)*, 2015.

[38] Yuxin Chen, Shervin Javdani, Amin Karbasi, James Andrew Bagnell, Siddhartha Srinivasa, and Andreas Krause. Submodular surrogates for value of information. In *Proc. Conference on Artificial Intelligence (AAAI)*, 2015.

[39] G. Graber, J. Balzer, S. Soatto, and T. Pock. Efficient minimal surface regularization of perspective depth maps in variational stereo. 2015.

[40] S. Soatto, J. Dong, and N. Karianakis. Visual scene representations: Scaling and occlusion in convolutional architectures. *Proc. of the ICLR Workshop; ArXiv: 1412.6607; UCLA CSD140023*, June 2015.

8 Supplementary Material

The following includes presentation material referenced in the main report.

8.1 3D Scene Modeling

The methodology for constructing 3D information of the scene is described in the material provided in this section. It was noted during the course of the program that the quality of the reconstruction, upon which accurate spatial reasoning depends, is impacted accuracy of the intrinsic parameters of the cameras and knowledge of the sensing geometry. The former was provided, but the latter was not. Consequently, state-of-the-art methods employing automatic detection of correspondences were utilized. The accuracy of these methods depends greatly on both the sensor geometry and the content of the scene. For some of the scenes, these were not adequate to yield acceptable performance and as a result, manual correspondences were needed.

garden sequence

2/24

objective

- support scene understanding
- 3-d representation
- *map*:
  - ground plane $(\boldsymbol{n}, d) \in \mathbb{S}^2 \times \mathbb{R}$
  - camera reference frames

$$\mathbf{F}_i = \begin{pmatrix} \mathbf{R}_i & \boldsymbol{t}_i \\ \mathbf{0} & 1 \end{pmatrix}, \quad (\mathbf{R}_i, \boldsymbol{t}_i) \in \mathrm{SE}(3)$$

3/24

agenda

1. 3-d reconstruction pipeline
   - correspondence
   - local pose estimation
   - global refinement
   - gauge fixation
2. analysis

4/24

1. reconstruction pipeline

correspondence

- standard approach:
  - interest point detection
  - SIFT descriptors
  - brute-force matching
  - homography $\mathbf{H}_{i_j}^{i_k} : \mathbb{P}^2 \to \mathbb{P}^2$
  - outlier rejection (RANSAC)
- if that fails:
  - manual correspondence
  - DLT

6/24

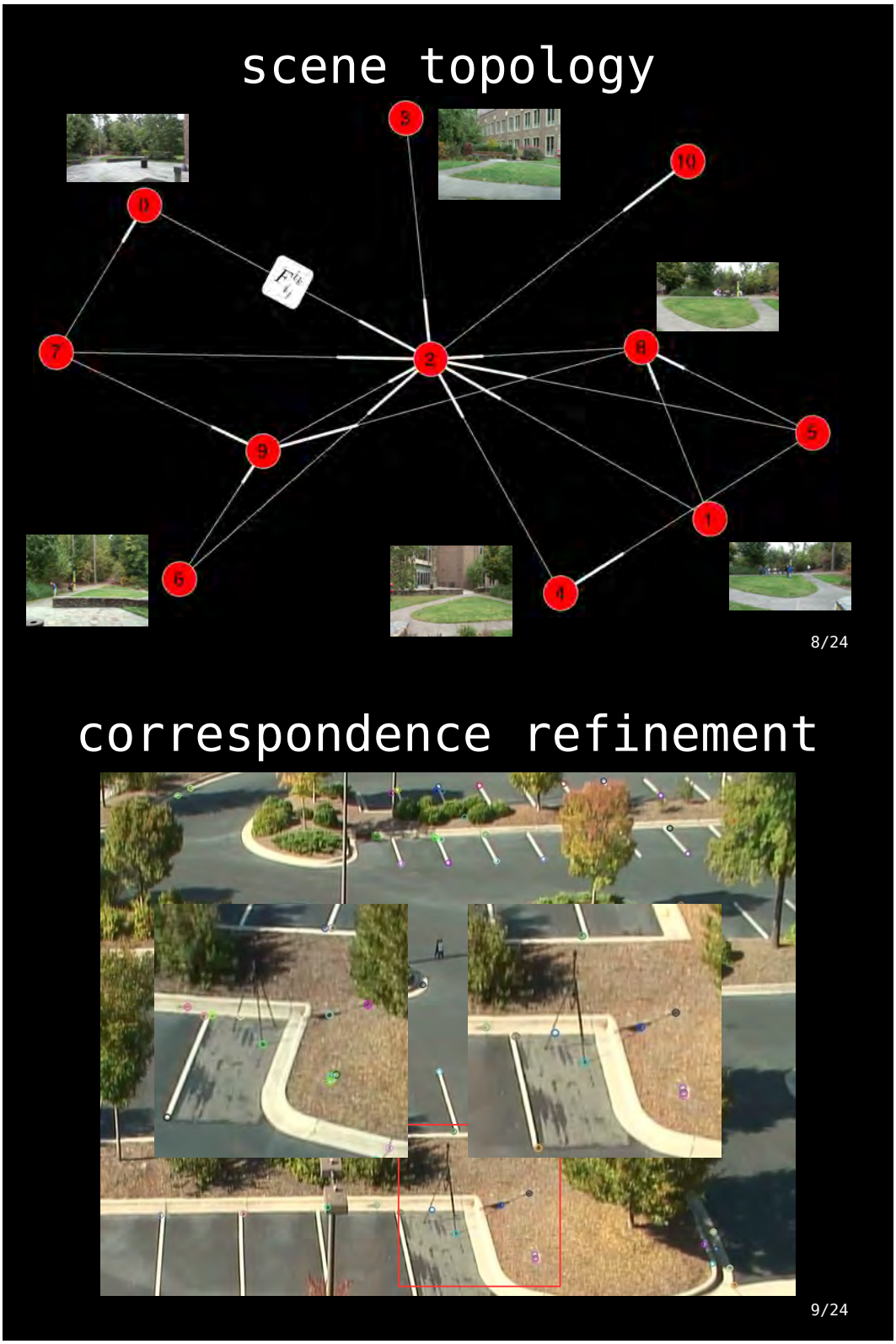
relative poses (local)

- projection matrices $\mathbf{K}_{i_j}, \mathbf{K}_{i_k} \in \mathbb{R}^{3\times 3}$
- Euclidean homography $\tilde{\mathbf{H}}_{i_j}^{i_k} = \mathbf{K}_{i_k}^{-1}\mathbf{H}_{i_j}^{i_k}\mathbf{K}_{i_j}$
- four decompositions of the form

$$\tilde{\mathbf{H}}_{i_j}^{i_k} = \mathbf{R}_{i_j}^{i_k} + \boldsymbol{n}_{i_j} \otimes \boldsymbol{t}_{i_j}^{i_k\top}$$

  (*twisted pair*)
- may need to pick solution manually

7/24



planar bundle adjustment

- objective function

$$f(\varphi, \theta, d, \mathbf{F}_i) = \sum_{i_j \neq i_k} \sum_{\substack{m \in \mathcal{I}(i_j) \\ n \in \mathcal{I}(i_k)}} \frac{1}{2} \| \boldsymbol{u}_n - \tilde{\mathbf{H}}_{i_j}^{i_k} \boldsymbol{u}_m \|^2$$

with

- correspondences $(\boldsymbol{u}_m, \boldsymbol{u}_n)$
- pairwise homography

$$\tilde{\mathbf{H}}_{i_j}^{i_k} = \mathbf{R}_{i_j}^{i_k} + \frac{1}{d_{i_j}} \boldsymbol{n}_{i_j} \otimes \boldsymbol{t}_{i_j}^{i_k \top} \qquad \mathbf{F}_{i_j}^{i_k} = \mathbf{F}_{i_k}^{-1} \mathbf{F}_{i_j}$$

planar bundle adjustment

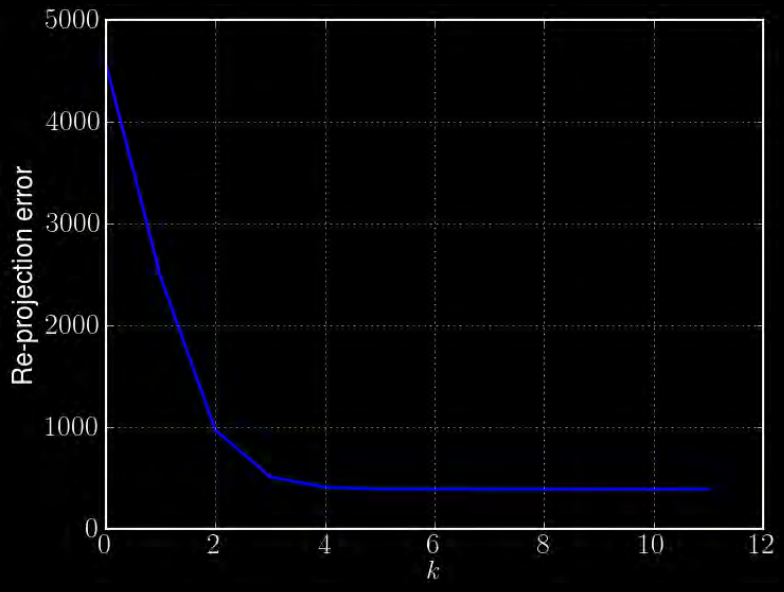
- "world" plane

$$\boldsymbol{n}(\varphi, \theta) = \begin{pmatrix} \sin\varphi\cos\theta \\ \sin\varphi\sin\theta \\ \cos\varphi \end{pmatrix} \qquad \boldsymbol{x}_0 = \begin{pmatrix} 0 \\ 0 \\ \frac{d}{\cos\varphi} \end{pmatrix}$$

- "local" plane

$$\boldsymbol{n}_{i_j} = \mathbf{R}_{i_j} \boldsymbol{n} \qquad d_{i_j} = \left\langle \boldsymbol{n}_{i_j}, \mathbf{R}_{i_j} \boldsymbol{x}_0 + \boldsymbol{t}_{i_j} \right\rangle$$

convergence

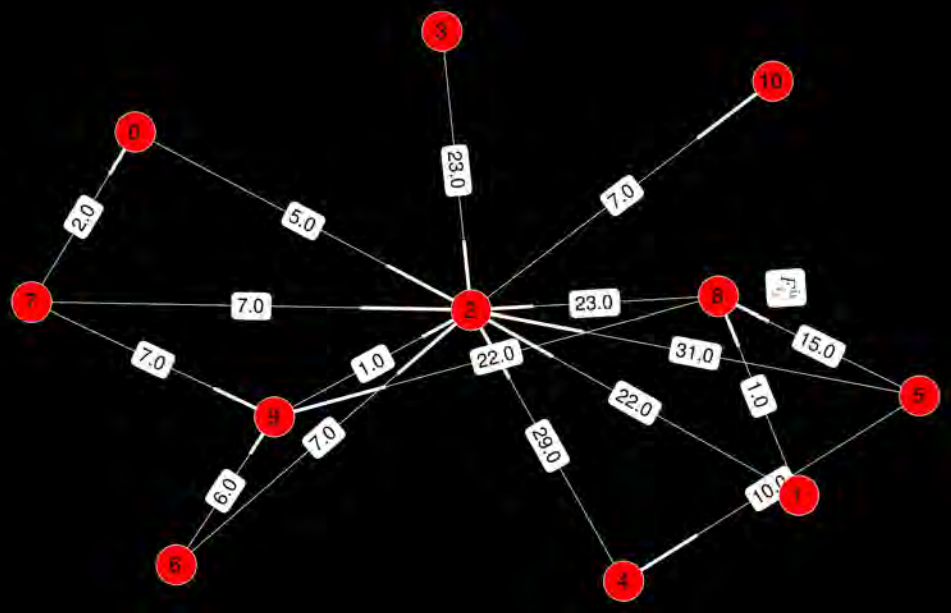


12/24

GPS coordinates

13/24

mutual distances



14/24

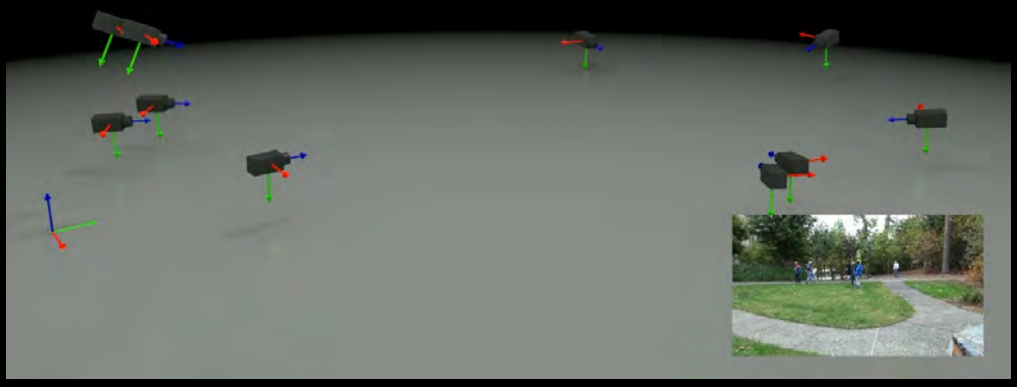
gauge fixation

- global scale is just the “units”
- *but*:
  - common to remote pairs
  - initialization of bundle adjustment
  - some predicates may depend on it
- post-mortem scale adjustment

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}} \sum_i \frac{1}{2} \|\boldsymbol{t}_i - \alpha \boldsymbol{t}_{\mathrm{GPS},i}\|^2$$

15/24

result



16/24

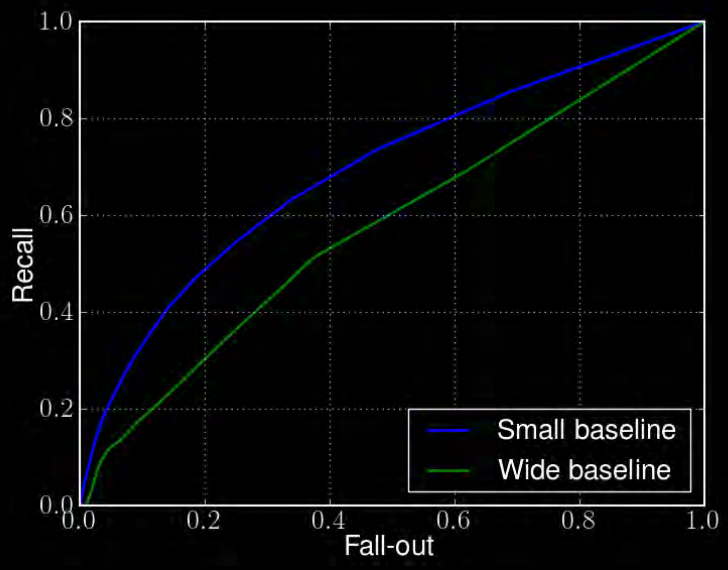
2. analysis

correspondence

- co-visible region
- texture
- shadows
- distortion

18/24

matching performance



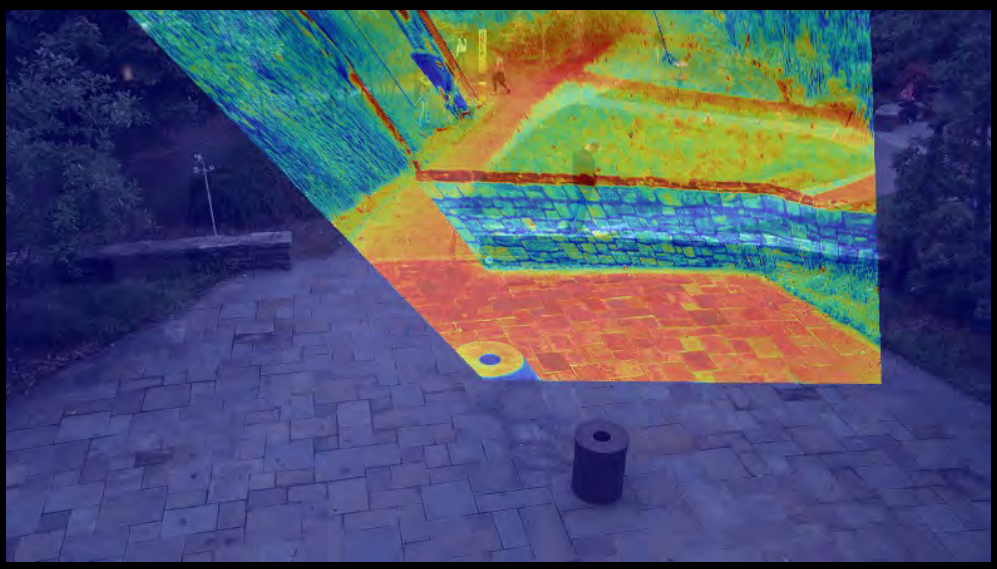
19/24

priors

- planarity assumption
- HUMINT
  - co-visibility
  - correspondence
- GPS data
  - unreliable elevation

20/24

planarity assumption

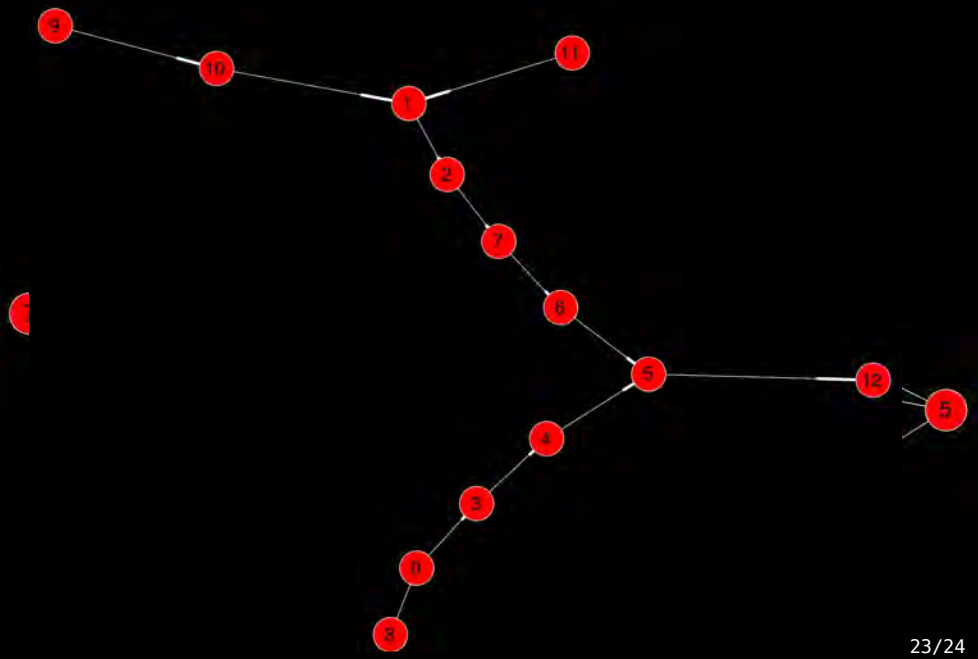


21/24

effective field of view

22/24

merits of BA?



23/24

moving cameras

24/24

Improved Scene Representation in MSEE Phase III

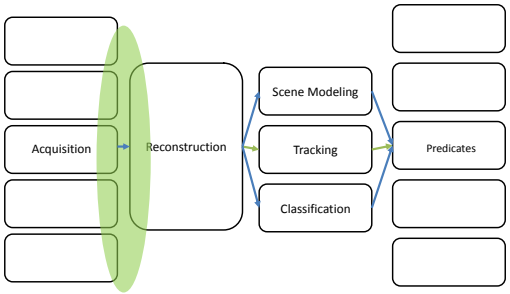
- Uncertainty in 3D representation.
- Multi-view tracking.
- More complete scene understanding for solid objects.
- More complete treatment of mobile cameras.

Uncertainty in 3D Reasoning

- Question: How uncertainty in 3D understanding affects predicates performance?
- Affects reconstruction, tracking, camera positions, object positions, etc.
- How to quantify - both in terms of algorithms, experiments, and ground truth data.

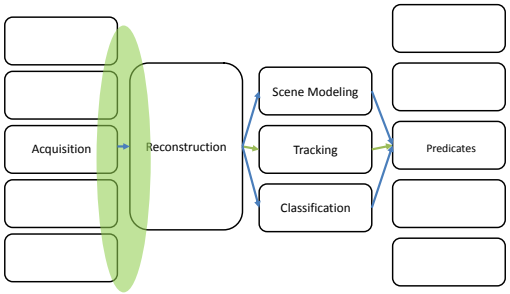
Uncertainty in 3D Reasoning

- 3D errors mostly created between image processing of acquired footage, image correspondence, and 3D reconstruction phases.
- 3D uncertainty propagates to the predicates.



Uncertainty in 3D Reasoning

- 3D understanding in our implementation is encapsulated by 3D wrapper functions:
  - Given a 2D point, fetch the 3D location.
  - Can include uncertainty estimates.



Uncertainty in 3D Reasoning

Many predicates benefit from 3D reasoning:

1. Clear line of sight, Occluding
2. Below, On, Closer, Farther, Together
3. Running, Sitting, Standing, Stopping, Turning, Walking, Crawling, Stationary, Entering, Exiting,

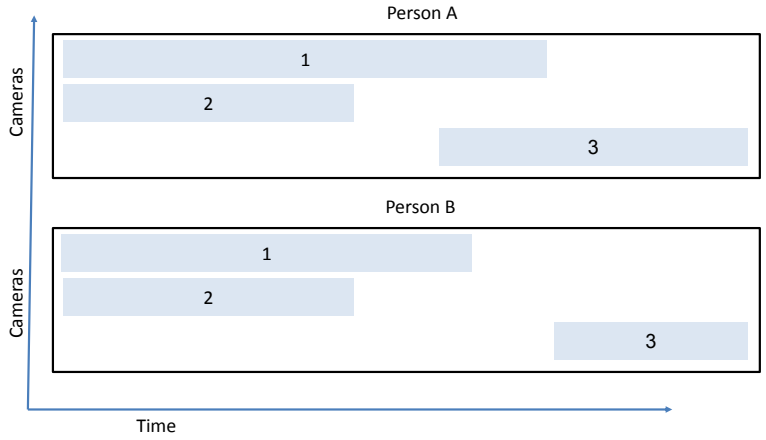
- Some of them are relative, and some are absolute.
- Tracking is key for most.

Uncertainty in 3D Reasoning

- The desired multiview tracking system should handle tracked objects in 0,1,2+ views.
- Should lend itself to analysis, prediction, and resource allocation.
- Some views are more informative for 3D location.
- Some views may be informative due to other data (appearance)
- Some view pairs are more informative.

Uncertainty in 3D Reasoning

Two examples of camera coverage - a good multiview tracker with uncertainty should cope with both!



Geometric Uncertainty in Predicates Computation

Several sources of for predicate errors related to object locations – among others:

- Segmentation errors
- Tracking errors
- 3D camera reconstruction errors
- 3D object reconstruction errors

Uncertainty in 3D Reasoning

Geometric Uncertainty Sources in Predicates Computation

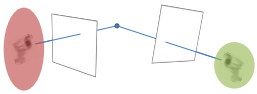
- Segmentation errors - wrong object boundary.

- Tracking errors - loss of tracking to background, switched tracks, tracks created from camera artifacts.

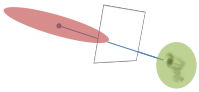
Uncertainty in 3D Reasoning

Geometric Uncertainty Sources in Predicates Computation

- 3D camera reconstruction errors - affect multiple objects.



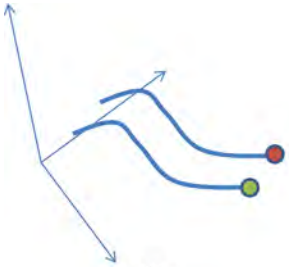
- 3D object reconstruction errors

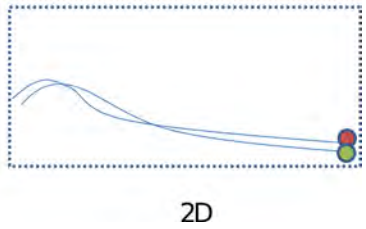


Uncertainty in 3D Reasoning

Geometric Uncertainty Sources in Predicates Computation

- In many cases, 2D/image-based predicates approximate 3D-based ones.
- They work better than 3D (as we tested..) when we do not have a good 3D scene model, and make some simplifying assumptions (i.e. implicit priors)

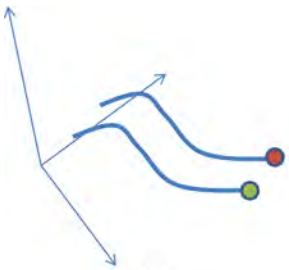


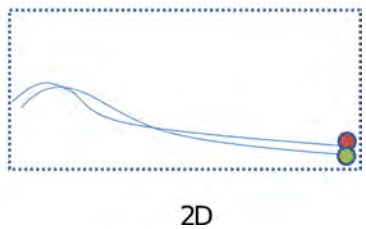


Uncertainty in 3D Reasoning

Geometric Uncertainty Sources in Predicates Computation

- Modeling 3D uncertainty would allow us to get the best of both worlds, by accounting both for error given 3D representation, and the representation error.
- Ample test data, where using all the viewpoints provides a stable 3D reconstruction/"ground truth", would allow us to quantify that.

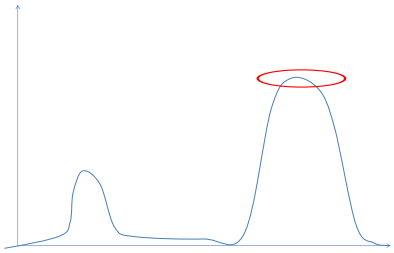




Reconstruction Error Sources

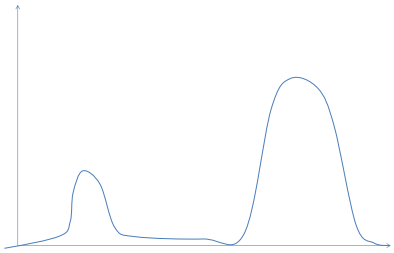
Mostly - errors introduced from features

- Small scale feature localization errors – these relate to noise/artifacts and inaccuracy in feature localization
- Correspondence error – relate to mismatches of feature points.
  - Correspondences are usually sampled in order to find the MAP solution (RANSAC). Correspondence errors often lead to reconstruction catastrophies.
  - Correspondences quality is a known question in comp. vision. with strong effect on the results.



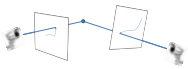
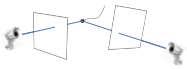
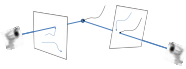
Reconstruction Error Sources

- Image correspondence errors - less common. Avoiding these depends on a strongly connected scene graph with many overlaps. Sensors GPS/location helps avoid some errors.
- Reconstruction packages (such as VSFM) provide some support for dictating image correspondences.

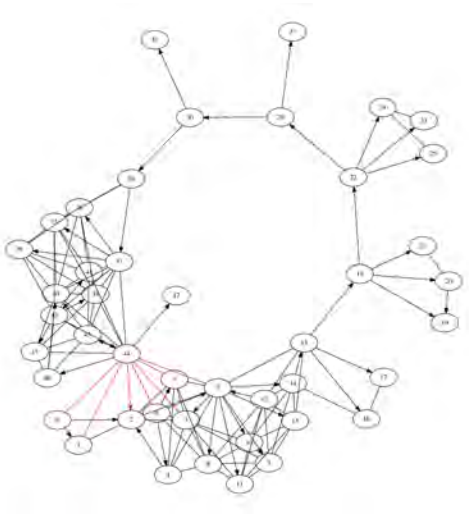


- Several approaches available for incorporating multiple views into tracking and classification
- In many cases, track loss can be minimized by combining hypotheses from multiple views.
- This includes both geometric reasoning (2D-3D association) and photometric reasoning
- Regardless of the specific method for dealing with the complexity of the space (Pruning/MHT, Sampling, DP/MAP, ..)

Incorporating 2D-3D association

- Track then reconstruct 
- Reconstruct then track 
- General association 
  With a 3D representation that explains 2D observations.
  - Note that generative models lend themselves for incorporating multi-sensor and multi-view data.
  - For efficiency reasons, we may favor 2D tracking, followed by 3D association.

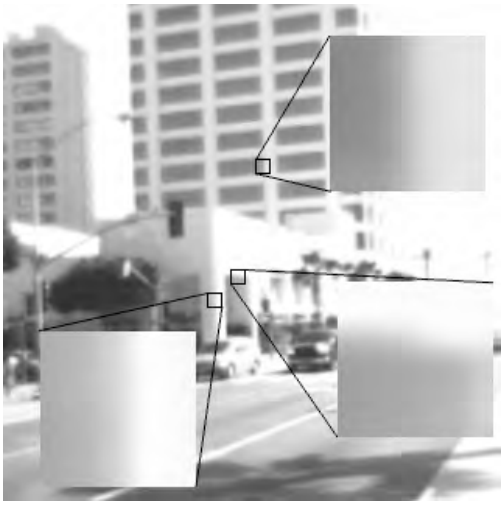
Scene representation



Scene graph

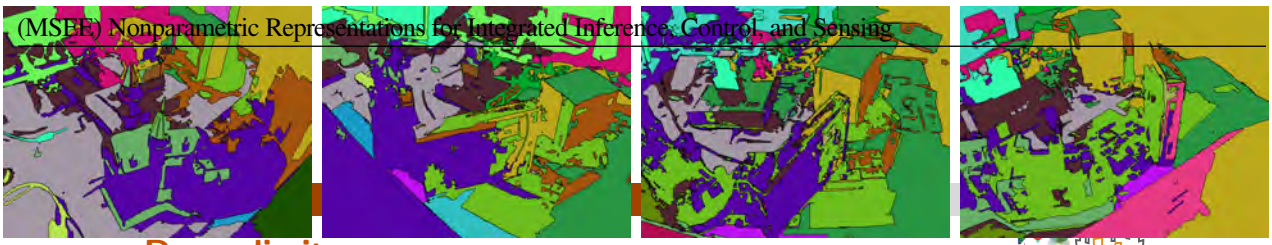
- nodes = locations/camera poses
- **collection of photometric attributes (features)**
- **edges = overlapping views**

Scene representation



View graph

- associated with a node of the scene graph
- nodes = geometric/photometric attribute
- edges connecting points belonging to the same surface



Dense limit

Fig. 3. *Our results on the 'schwanstein' sequence (top) compared to [?] (bottom)*

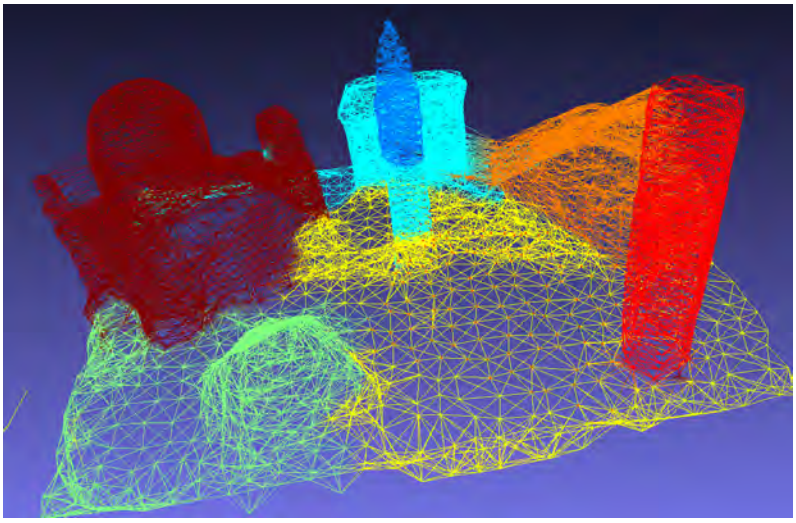


Fig. 4 ***MPO-RAR***

Semantic representation

- Scene representation partitioned into objects.
- Objects project onto image, providing pixel-level video object segmentation.

3.2

4 C

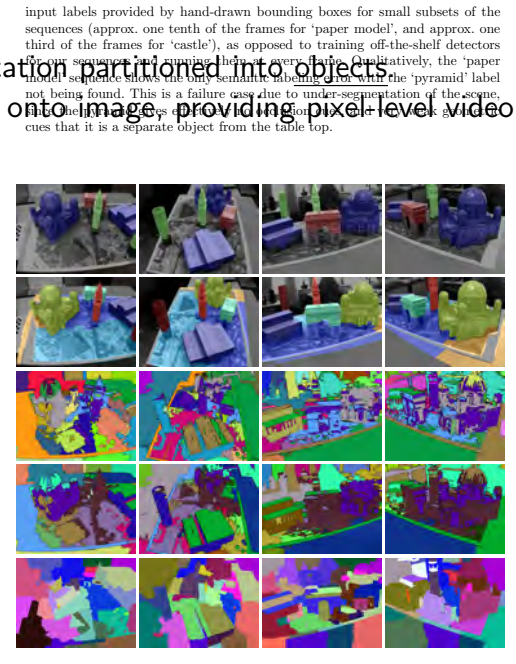
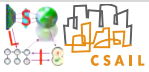


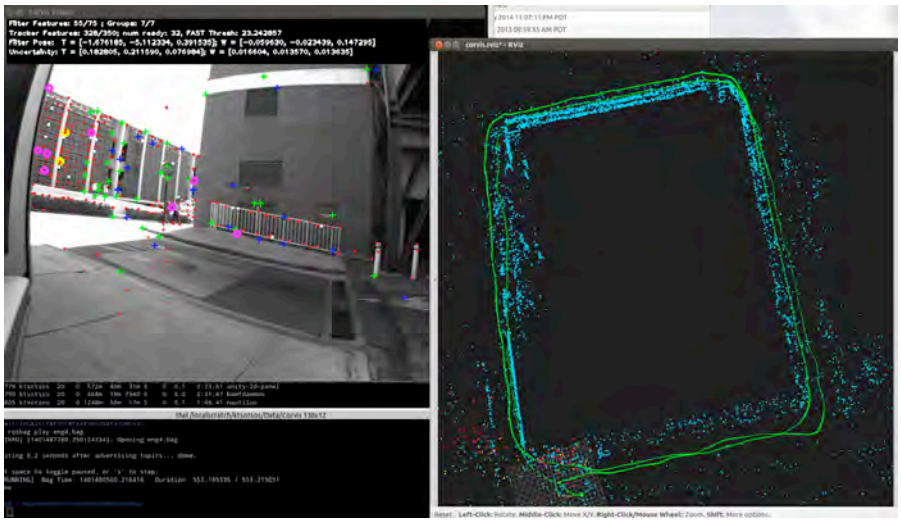
Fig. 5. *Results on 'paper model' sequence. Our output semantic labeling (top) with output object segmentation (second), video segmentation results from [52] tuned for temporal consistency (third), results from [52] tuned for similar number of segments to our results (fourth), single image segmentation results from [55] tuned for similar number of segments.*

Representation inference

- "MAP" approach:
  - Geometry reconstructed through one of many variants of bundle adjustment.
  - No topology, no uncertainty estimate in the reconstruction
  - This was the approach adopted in first evaluation (see below).
- Bayesian approach:
  - Geometry and local photometry estimated as part of a filtering process.
  - Allows incorporation of inertial sensing priors.
  - Benefits from continuous camera trajectories (see below).
  - Provides uncertainty estimates on pose as well as scene geometry.

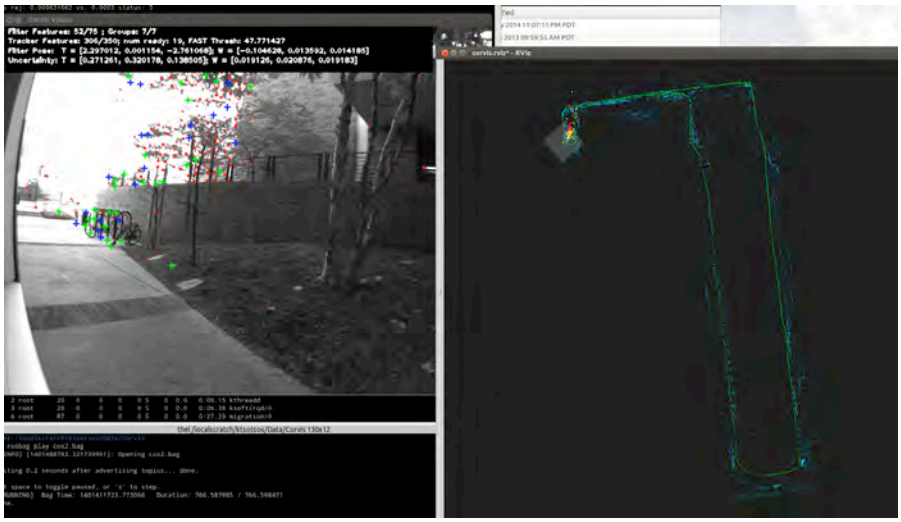
Corvis





Boelter Hall loop I

Corvis



Boelter Hall loop II

3D scene understanding in Phase 3 – summary

- Uncertainty quantification in a point-estimate setting.
- Incorporating 3D reasoning into tracking.
- Partition the scene into objects/primitives (e.g. groups of points and their connectivity)
- Testing of filtering approach provided sequences are given with accurately synchornized video taken from a moving platform (e.g. quadrotor) with no rolling shutter artifacts

8.2 Object Tracking

As part of pre-processing the MIT design tracks all movers, storing the results in a database. Both the location (within the sensor view) and the boundary of the object are computed. The tracker was partially developed under the MSEE program and implements layered tracking, adaptive appearance models, and occlusion reasoning.

Tracking: Why Do We Need Tracking?

Queries over time windows ⇒ Need data association across frames

Example: how many cars appear in the sequence above?
To report the right answer (one), we need to know it is the same car.

The Tracker: A Layered Representation

1. $N + 1$ classes: background + N objects.
2. Object j is represented as a binary mask, denoted M_j.
3. Depth ordering: Z is permutation of $\{1, \ldots, N\}$. E.g. if $N = 4$ and $Z = (1, 3, 4, 2)$, then object 2 is the closest to the camera.
4. $L(x) \in \{0, 1, \ldots, N\}$: pixel label at location x.
   If $\max_{j \in \{1,\ldots,N\}} M_j^t(x) = 0$ then it is background: $L(x) = 0$. Otherwise,

$$L(x) = \underset{\{j : M_j(x) = 1\}}{\arg\max} \; Z(j)$$

(video)

The Tracker: Probabilistic Modeling

1. Binary maps updates: $M_j^t(x)$ given by

$$\underset{M_j^t(x)\in\{0,1\}}{\arg\max}\ \Pr(M_j^t(x)|I(x), \overbrace{\underbrace{Z}_{\text{ordering}}, \underbrace{A}_{\text{appearance}}, \underbrace{v}_{\text{velocity}}}^{\text{latent variables}}, M_j^{t-1}(x))$$

2. Appearance:

   models: $p(I(x)|A^{(j)}, L(x)=j)$ parameters: $A=(A^{(0)}, A^{(1)}, \ldots, A^{(N)})$

3. N velocities: $v=(v^{(1)}, \ldots, v^{(N)})$
4. Depth ordering: Z

Appearance

1. The parameters: $A=(A^{(0)}, A^{(1)}, \ldots, A^{(N)})$
2. (A) A pixel-wise background model:

$$A^{(0)}=A^{(0)}(x) \qquad p(I(x)|A^{(j)}, L(x)=0) \sim \mathcal{N}(\overbrace{\mu^{(0)}(x), \Sigma^{(0)}(x)}^{A^{(0)}(x)})$$

   (B) Each object has one GMM model:

$$A^{(j)} \overset{j\geq 0}{=} \{w_k^{(j)}, \mu_k^{(j)}, \Sigma_k^{(j)})\}_{k=1}^K$$

$$p(I(x)|A^{(j)}, L(x)=j) \sim \sum_{k=1}^K w_k^{(j)} \mathcal{N}(\mu_k^{(j)}, \Sigma_k^{(j)})$$

Appearance: Initial Background Model

Temporal Median: $m(x) = \text{median}(I_{t=1}(x), I_{t=2}(x), \ldots)$
$\mu^{(0)}(x) \leftarrow m(x)$
$\Sigma^{(0)}(x) \leftarrow \frac{1}{\#\text{frames}-1} \sum_t (I_t(x) - m(x))^T (I_t(x) - m(x))$

Zooming in:

Velocity

Object velocity implies a per-pixel prior

1. $v_{t-1}^{(j)}$: velocity of object i between frame $t-2$ and frame $t-1$.
2. Applying $v_{t-1}^{(j)}$ to M_j^{t-1} yields a new mask at frame t.
3. Distances from the new mask are used to (inversely) weight the pixels.
4. Pixels far from the new mask are unlikely to be classified as object j at frame t

Ordering: θ_{ord}

1. Explicit modeling of the ordering helps to deal with occlusions.
2. Z = a permutation of $\{1, \ldots, N\}$
3. $$p(\underbrace{I}_{\text{color}} | \underbrace{A}_{\text{appearance}}, \underbrace{Z}_{\text{ordering}}, \underbrace{M_1, \ldots, M_N}_{\text{object masks}})$$
   Propose Z'; if $p(I|A, Z', \{M_j\}_{j=1}^N) > p(I|A, Z, \{M_j\}_{j=1}^N)$ then $Z \leftarrow Z'$.
4. There are $N!$ options – but we only need to consider a subset of these: If objects don't overlap, their depth ordering doesn't matter

Parameter Updates

- L^t is determined by binary masks and the ordering.
- Given I, L^t can estimate new A.
  Then use a convex combination with previous estimates.
  E.g., $A = \alpha \times A^{\text{old}} + (1 - \alpha) \times A^{\text{new}}$, where $\alpha = 0.1$.
- Given M_j^{t-1} and M_j^t can estimate new velocity.

Changing the Number of Objects

Use a simple heuristic to establish N

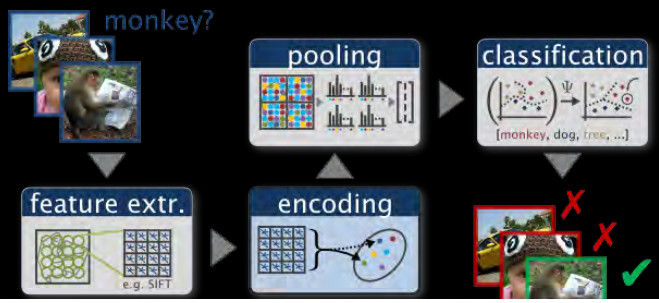
1. An object can "die" if doesn't have enough image evidence.
2. For creating new objects, we consider, among the pixels labeled as background, the connected components of low-likelihood pixels.

8.3 Object Classification

As part of pre-processing the MIT all movers and static objects are classified using a variant of Caffe adapted to the MSEE object hierarchy. The implementation provided by ICSI (co-PI Darrell) did not fully implement the hierarchy, but nevertheless provided reasonable performance on many objects of interest. One impact on performance is that only the highest scoring class was maintained for each object. As such, errors in the use of the classfier had an undue impact on system performance. More robust performance would be obtained if a full or partial distribution were maintained as part of the pre-processing. This is feasible, but would complicate query processing owing to the increased combinatorial complexity. Consequently, the simple approach was chosen for phase 2.

Traditional Vision Models…

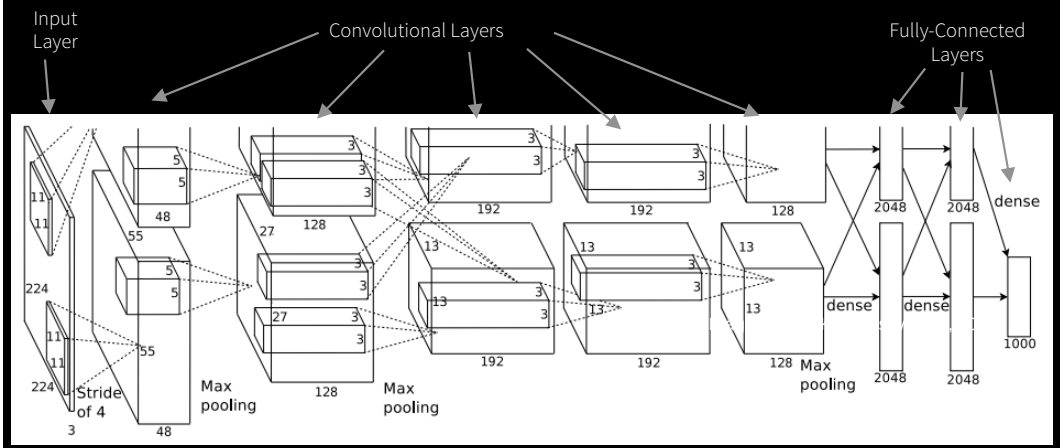
SIFT-VQ-BOW



Scanning Window HOG

Convolve-Quantize-Pool → [Convolve-Quantize-Pool]

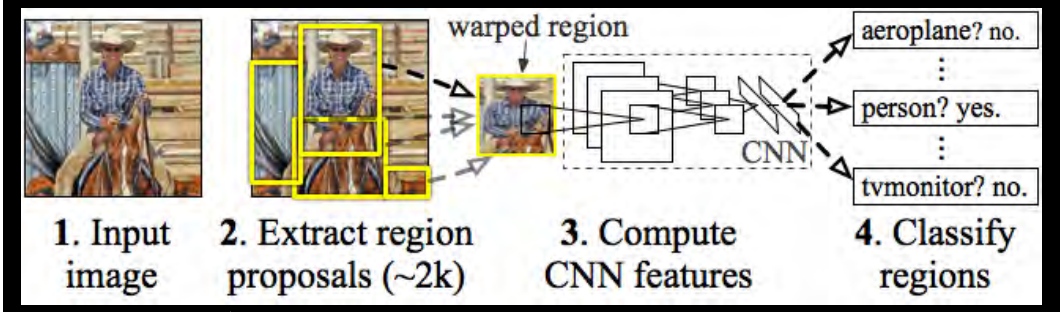
…now, CNN ILSVRC Architecture:

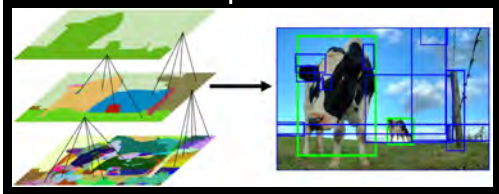


Convolve-Quantize-Pool → [Convolve-Quantize-Pool] → [[Convolve-Quantize-Pool]]→ …

Fukushima's **Neocognitron** 1974-82; LeCun's **LeNet**, 1989;
Krizhevsky, A., Sutskever, I., and Hinton., G. E. **ImageNet Classification with Deep Convolutional Neural Networks**. In *Proc. NIPS*, 2012.

"Regions with CNN features" (R-CNN)





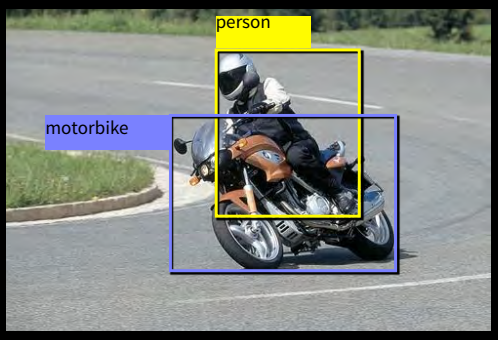
(e.g., "selective search")

(With a few minor tweaks: semantic segmentation)

Object detection

{ airplane, bird, motorbike, person, sofa }

Input



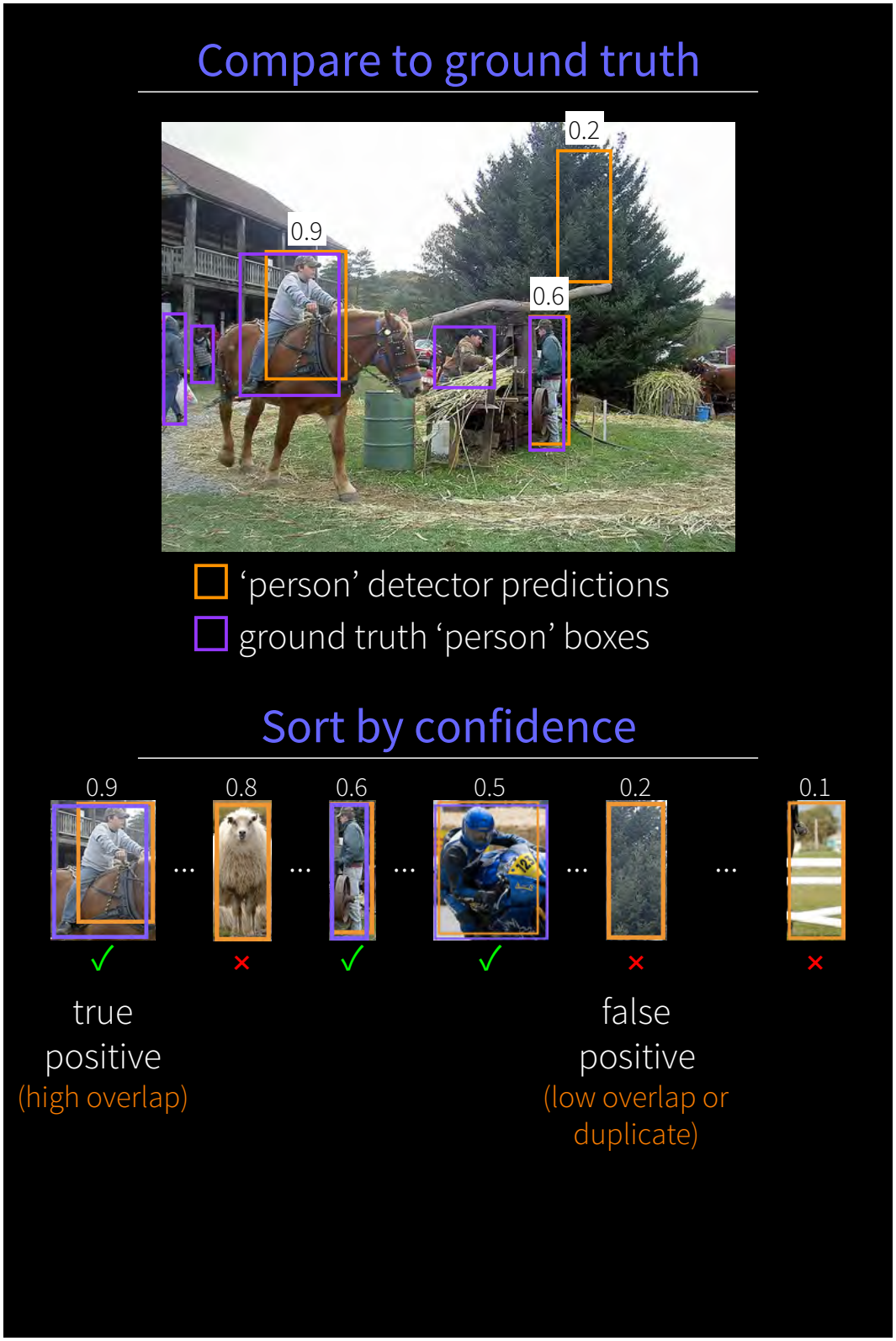
Desired output

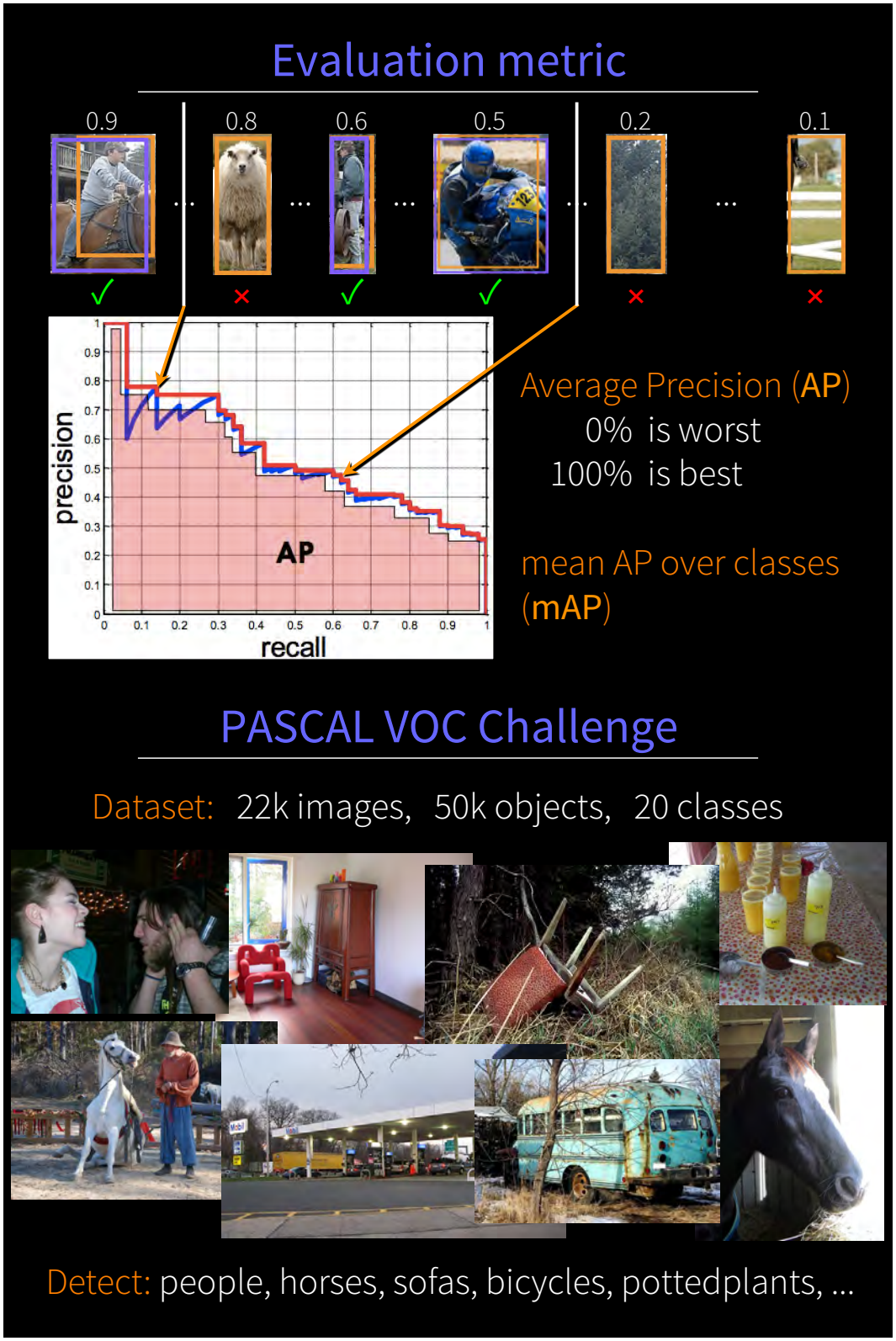
Evaluating a detector

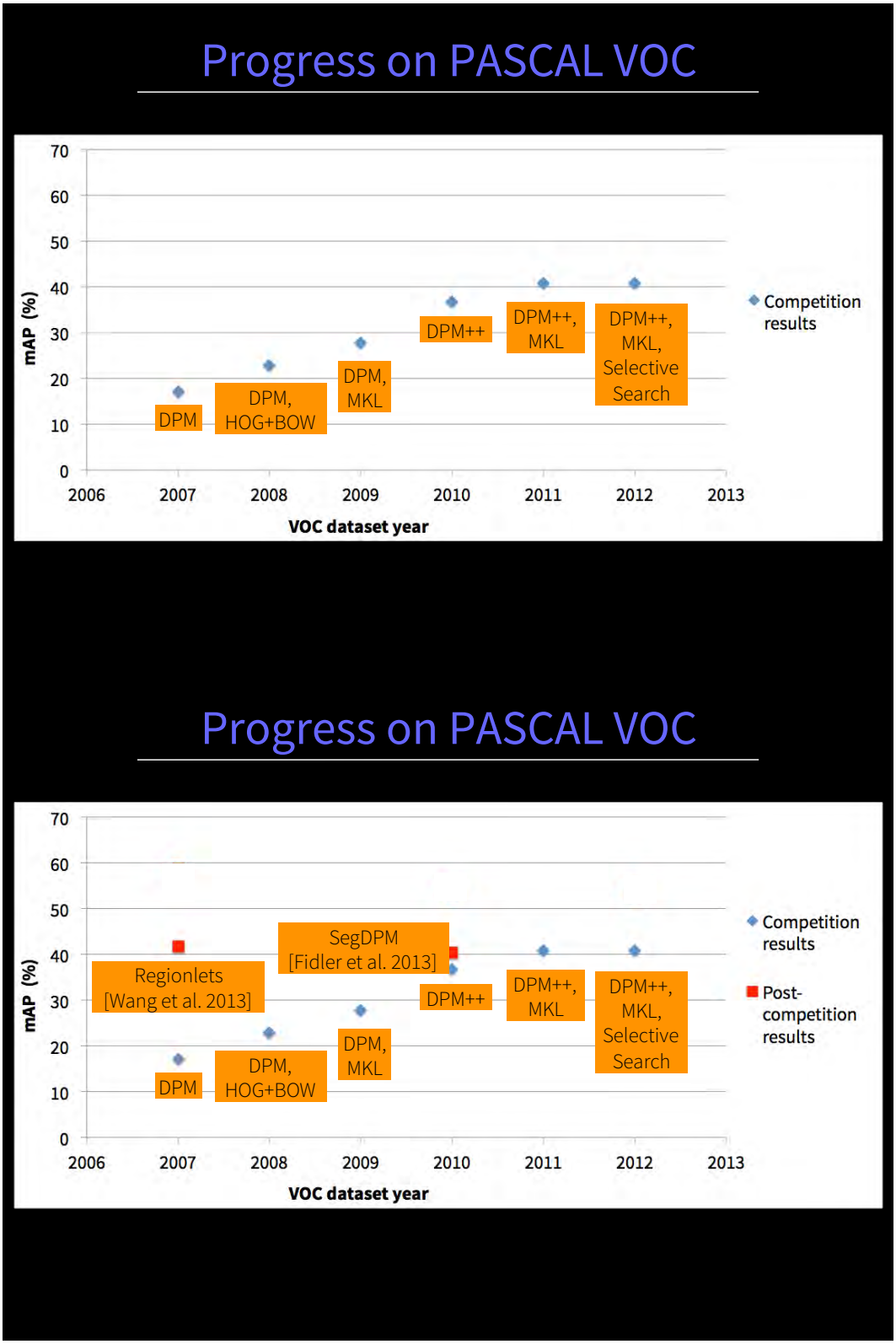
Test image (previously unseen)

First detection …

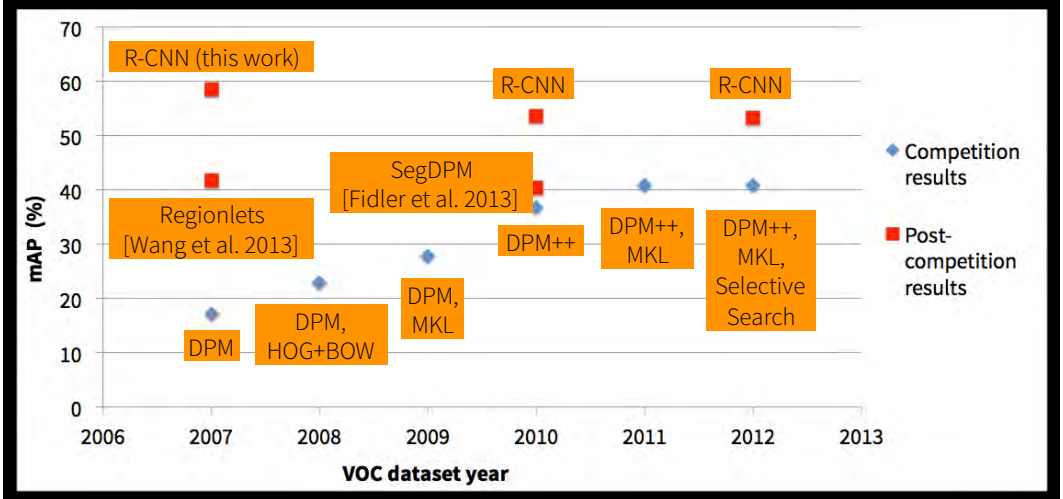
☐ ‘person’ detector predictions







Progress on PASCAL VOC



ImageNet LSVR Challenge

– 1000 classes
(vs. 20)

– 1.2 million training images
(vs. 10k)

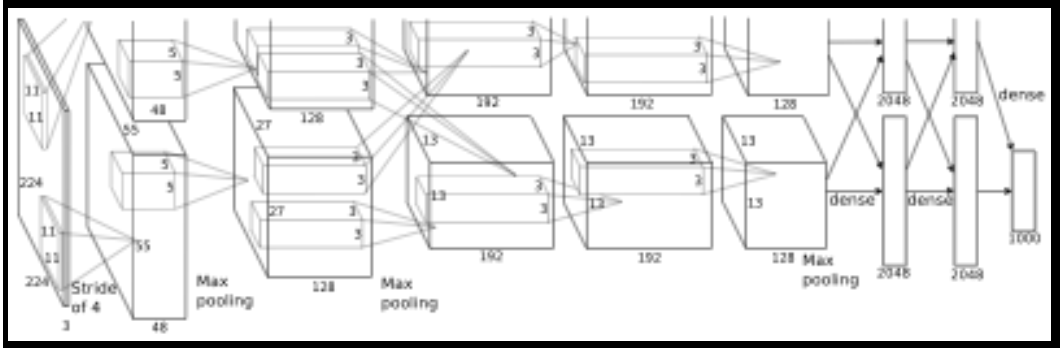
– Image classification
(not detection)

bus anywhere?

[Deng et al. CVPR'09]

Multi-layer feature learning

"SuperVision" Convolutional Neural Network (CNN)



input ← 5 convolutional layers → fully connected

ImageNet Classification with Deep Convolutional Neural Networks.
Krizhevsky, Sutskever, Hinton. NIPS 2012.

cf. LeCun et al. Neural Comp. '89 & Proc. of the IEEE '98

Impressive ImageNet results!

1000-way image classification

	Top-5 error
Fisher Vectors (ISI)	26.2%
5 SuperVision CNNs	16.4%
7 SuperVision CNNs	**15.3%**

now: 12%

metric: better)

But... does it generalize to other datasets and tasks?

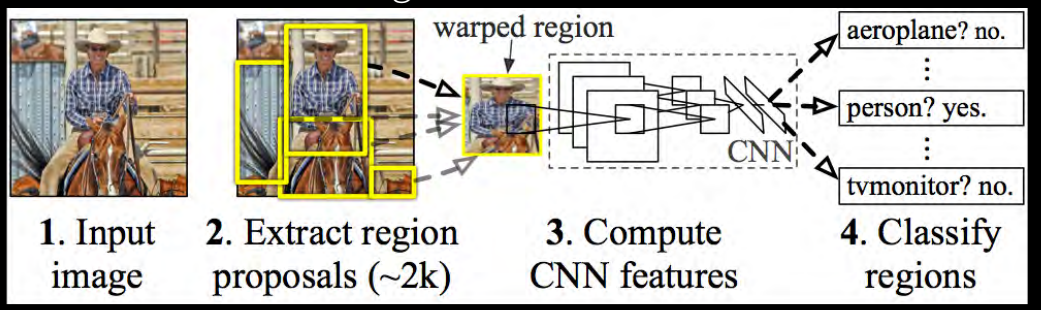
Spirited debate at ECCV 2012

Objective

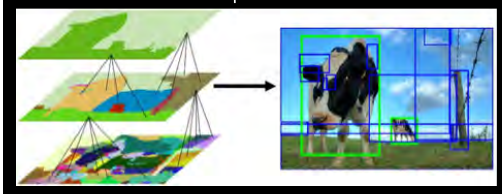
Can the SuperVision CNN detect objects?

Proposed system

R-CNN: "Regions with CNN features"



[Girshick, Donahue, Darrell, Malik to appear in CVPR'14]



"selective search" [van de Sande et al. 2011]

R-CNN results on PASCAL

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2012)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%

metric: mean average precision (higher is better)

R-CNN results on PASCAL

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2012)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%
R-CNN	54.2%	50.2%
R-CNN + bbox regression	**58.5%**	**53.7%**

metric: mean average precision (higher is better)

ImageNet detection (ILSVRC2013)

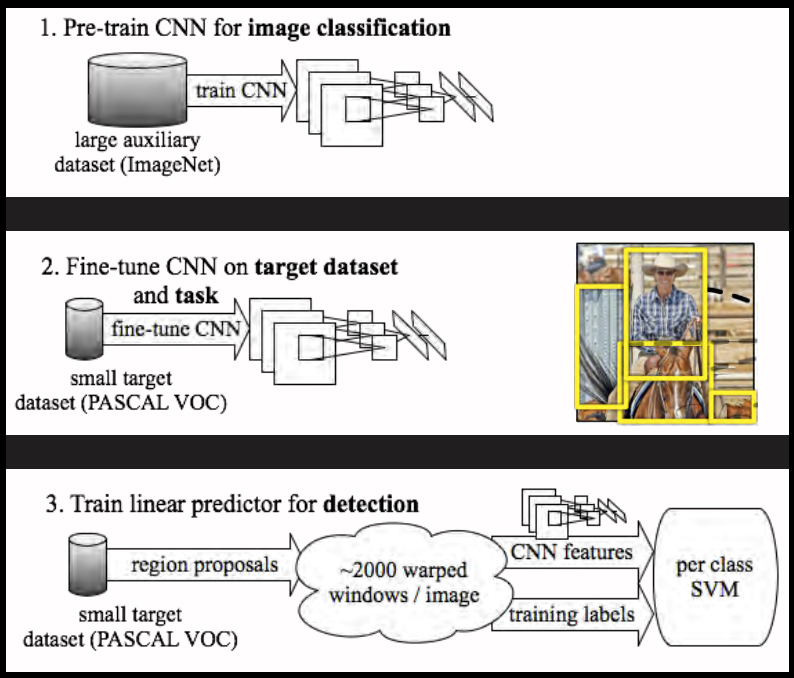
R-CNN and OverFeat

OverFeat [Sermanet et al. 2014]
- – developed using ILSVRC2013
- – tested on ILSVRC2013: s-o-t-a
- – no results on PASCAL VOC

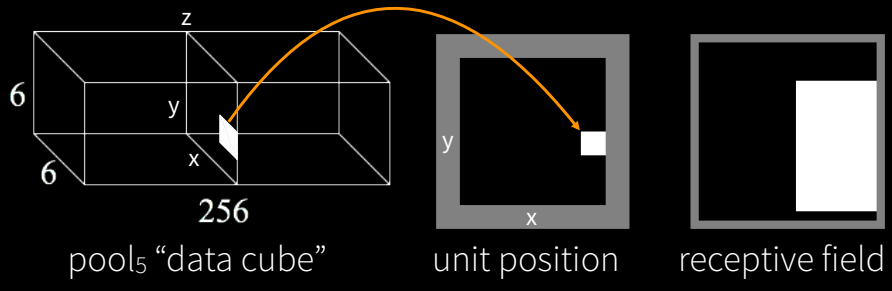
R-CNN [Girshick et al. 2014]
- – developed using PASCAL VOC
- – tested on PASCAL VOC: s-o-t-a
- – no results on ILSVRC2013

No apples-to-apples comparison

R-CNN detector training

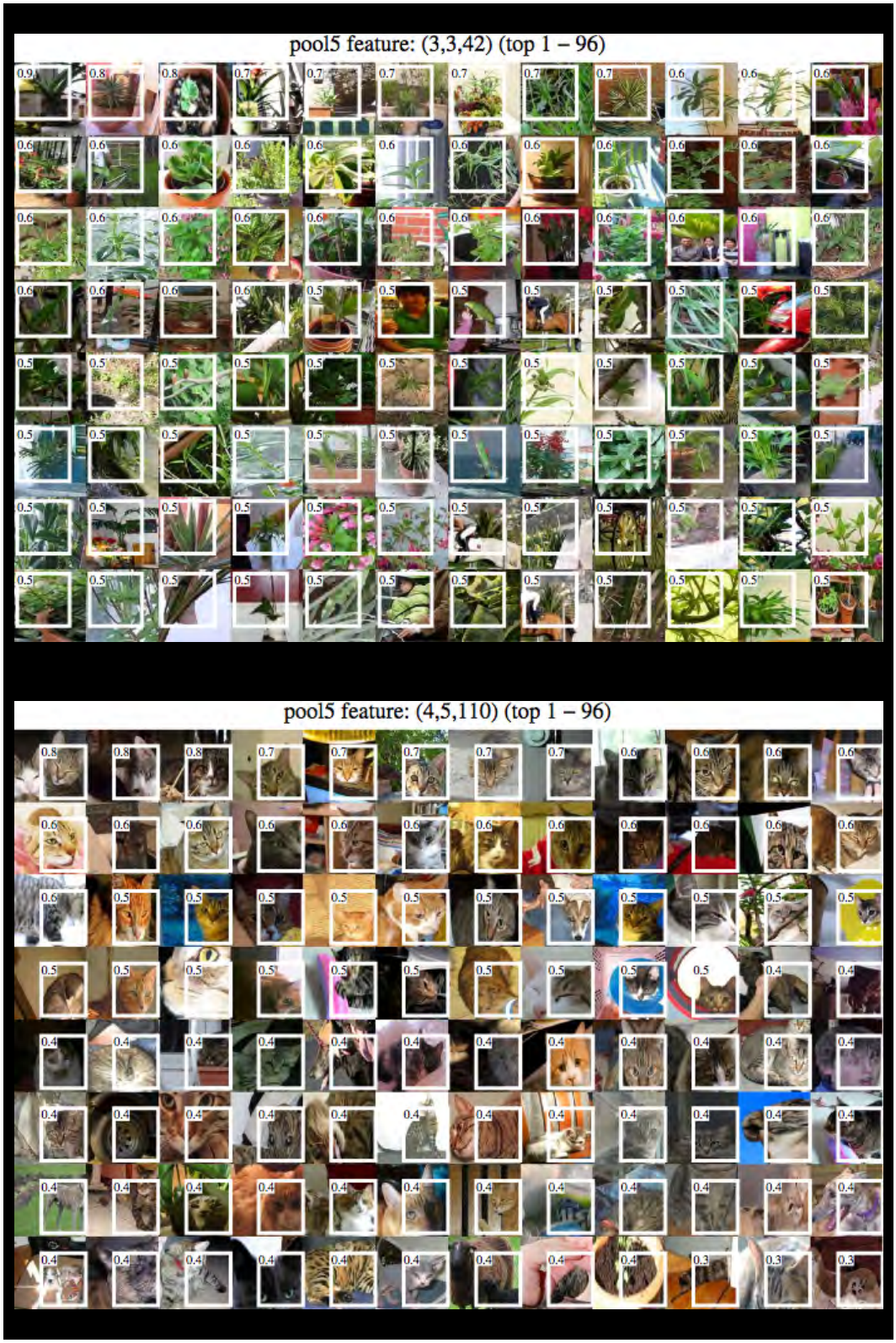


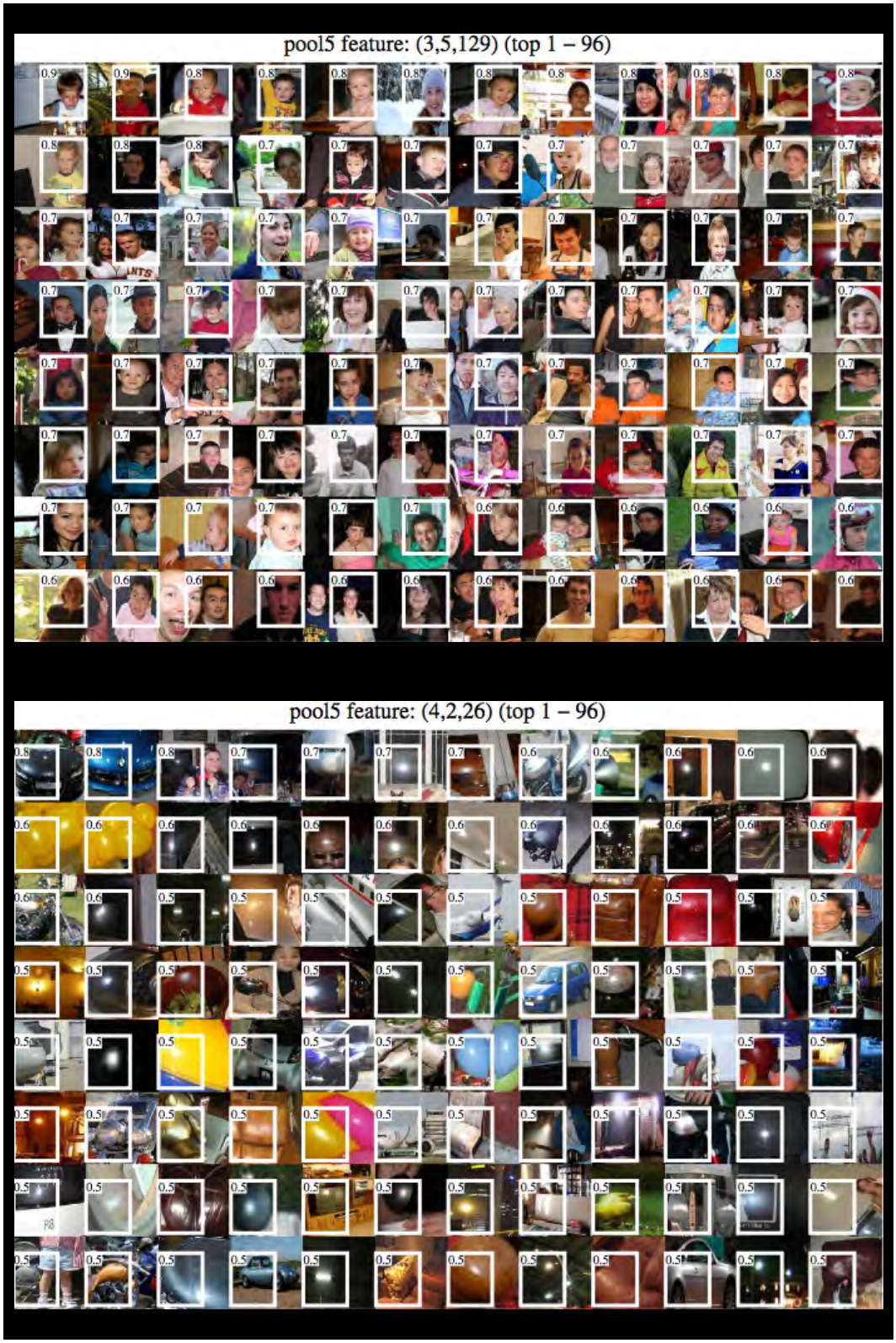
What did the network learn?



pool$_5$ "data cube" unit position receptive field

Visualize pool$_5$ units





MSEE Phase 2 RCNN Component

- Leverage ImageNet-derived representation (from Imagenet-1K)
- Use all ImageNet classes to train new class on top of R-CNN model.
- Find Nearest class in Imagenet to MSEE ontology.
- Fixed apriori mapping for P2
- Significant limitations: no Person subclasses

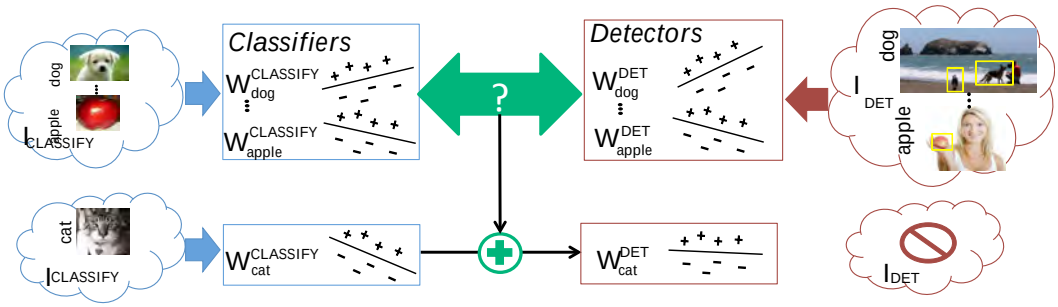
MSEE Phase 3 RCNN Plans

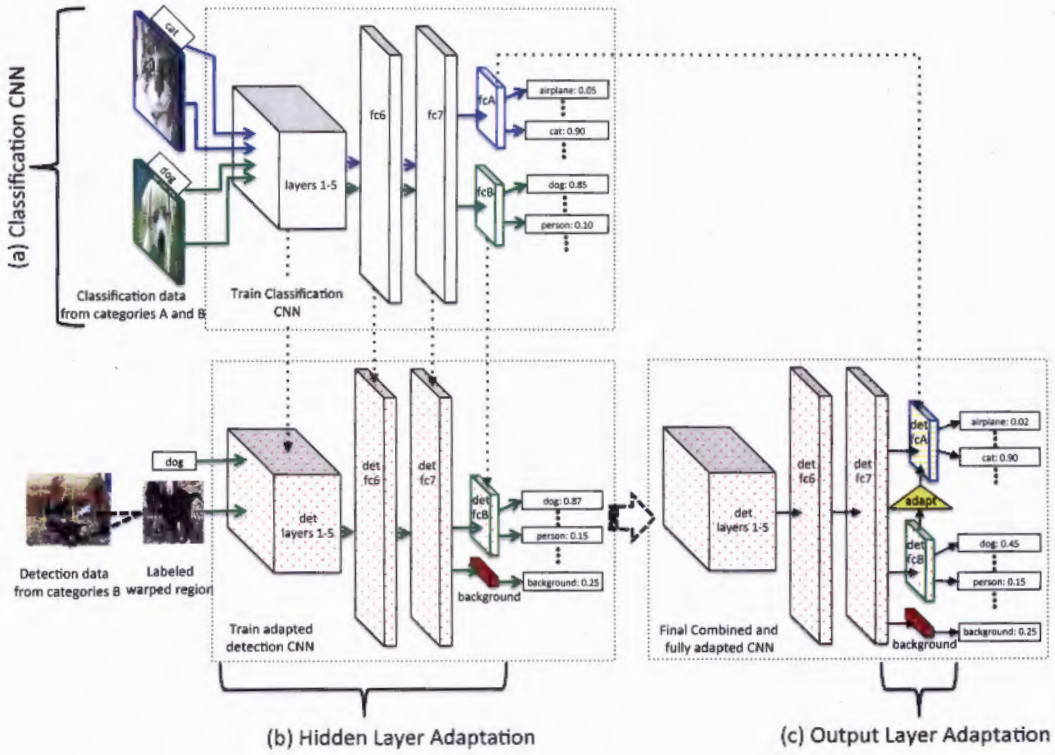
- Exploit adaptation (2 NIPS 2014 papers in review)
- Take in-domain examples as well as ImageNet training data
- Add new data for explicit person and vehicle subclass
- Fast training on the fly
- Tree-based loss for reasoning within hierarchy

4. Detection as Adaptation: Generalizing to new categories…

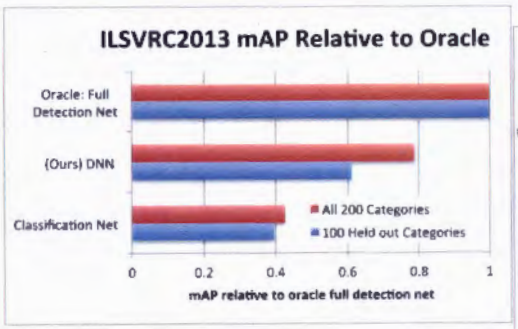
- NIPS 2014, in review.
- (To be released on arXiv, ca. July 2014)

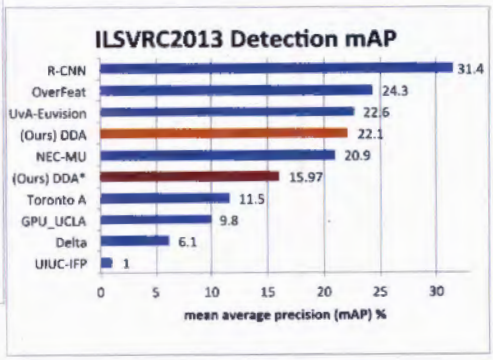
Detection as Adaptation





Results





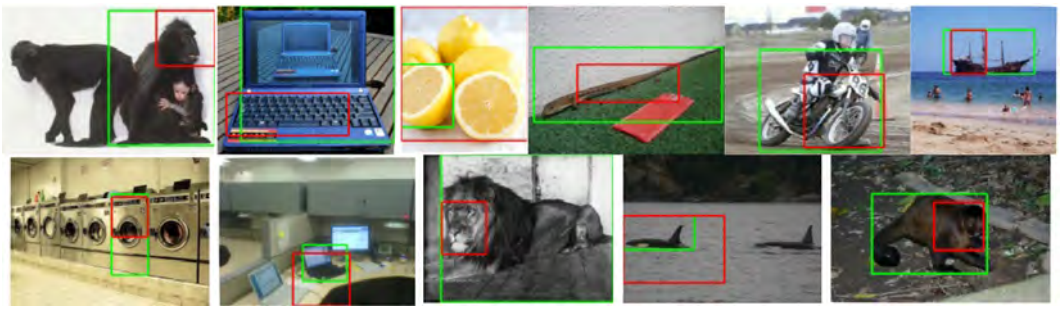
Ablation results

Detection Adaptation Layers	Output Layer Adaptation	mAP Trained 100 Categories	mAP Held-out 100 Categories	mAP All 200 Categories
No Adapt (Classification Network)		12.63	10.31	11.90
fc_{bgrnd}	-	14.93	12.22	13.60
fc_{bgrnd},fc_6	-	24.72	13.72	19.20
fc_{bgrnd},fc_7	-	23.41	14.57	19.00
fc_{bgrnd},fc_B	-	18.04	11.74	14.90
fc_{bgrnd},fc_6,fc_7	-	25.78	14.20	20.00
fc_{bgrnd},fc_6,fc_7,fc_B	-	26.33	14.42	20.40
fc_{bgrnd},layers1-7,fc_B	-	27.81	15.85	21.83
fc_{bgrnd},layers1-7,fc_B	Avg NN (k=5)	**28.12**	**15.97**	22.05
fc_{bgrnd},layers1-7,fc_B	Avg NN (k=100)	27.91	15.96	21.94
Oracle: Full Detection Network		29.72	26.25	28.00

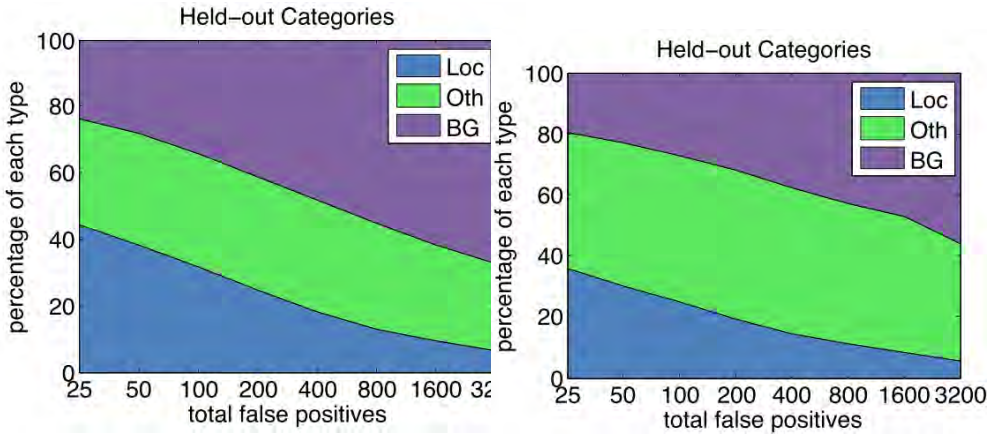
Table 1: Ablation study for the pieces of DNN. We consider removing different pieces of our algorithm to determine which pieces are essential. We consider training with the first 100 (alphabetically) categories of the ILSVRC2013 detection validation set (on val1) and report mean average precision (mAP) over the 100 trained on and 100 held out categories (on val2). We find the best improvement is from fine-tuning all convolutional fully connected layers and using output layer adaptation.

Near misses of adapted models

Localization is improved



Localization is improved



Detection Summary (RCNN vs DPM)

- ~150% improvement in raw performance training from ImageNet alone
- ~50% improvement in raw performance when training from 1—3 examples in domain